# közlemények

20/1978



MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest

MAGYAR TUDOMÁNYOS AKADÉMIA
SZÁMITÁSTECHNIKAI ÉS AUTOMATIZÁLÁSI KUTATÓ INTÉZETE

# KÖZLEMÉNYEK

1978. ÁPRILIS

Felelős kiadó:

DR VÁMOS TIBOR

igazgató

ISBN 963 311 061 0
ISSN 0133-7459

Technikai szerkesztő:
Solt Jánosné

# TARTALOMJEGYZÉK

CONTENTS

Procedings of the Computer and Automation Institute
Hungarian Academy of Sciences
Vol. 20.

# СОДЕРЖАНИЕ

Труды Исследовательского Института
Вычислительной Техники и Автоматизации
Венгерской Академии Наук

Выпуск 20.

MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények** 20/1978.

# AZ M HATÁRÉRTÉK – LOGIKA MONOTON OSZTÁLYAIRÓL

Gárdos Éva

## 1. § Bevezetés

A határérték logika fogalmát Sz.V. Jablonszkij vezette be [13] 1958-ban. Ennek a fogalom nak a szükségessége a véges logikák [12] és a végtelen értékű [14] tanulmányozásánál merült fel. Ugyanis a végtelen értékű logikák kontinuum sok függvényt tartalmaznak, igy ezen logikák kezelése meglehetősen nehézkes. Ezért vetődött fel egy olyan logika fogalom szükségessége, amely megszámlálható sok függvényt tartalmaz és bizonyos értelemben modelljének lehetne tekinteni minden *k*-értékű logikának. Ellentétben a többértékű és végtelen értékű logikákkal, a határérték-logika modelljének sok realizációja van, pontosabban kontinuum sok páronként nem izomorf határérték-logika létezik [1].

A vizsgálatok elsősorban az ekvivalencia osztályokra irányulnak és minden egyes parciális rendezés alkalmával az is az érdeklődés középpontjában van, hogy létezik-e minimális és maximális elem. A legismertebb és legegyszerűbb ekvivalencia az izomorfizmus. A [6, 8, 13] munkákból ismerets, hogy létezik kontinuum sok páronként nem izomorf határérték-logika.

Bizonyitást nyert [1], hogy az ismert algebrai rendezések mellett nincs maximális és minimális elem. A [1, 11] dolgozatban logikai uton bevezetett parciális rendezés a határérték-logikák halmazát szintén ekvivalanes osztályokra bontják fel és ilyen parciális rendezés mellett már léteznek maximális és minimális határérték-logikák. Maximális abban az értelemben, hogy tartalmaz minden más határérték-logikát, minimális abban az értelemben, hogy benne van minden más határérték-logikában.

Sz.V. Jablonszkij [12]-ben megvizsgálta a *k*-értékű logika majdnem teljes osztályait. Ezeket az osztályokat részben ő és még sokan mások [4, 5, 7, 11, 12] tanulmányozták.

Dolgozatnak célja az általa megvizsgált osztályok egyes reprezentánsainak vizsgálata egy $M$ maximális határérték-logikában a [12]-ben megadott módszer segitségével.

A *k*-értékű logikában véges sok majdnem teljes osztály van. Dolgozatunkban bebizonyitjuk, hogy az általunk vizsgált határérték-logikában kontinuum sok monoton majdnem teljes osztály van. Sikerült általánositanunk a Jablonszkij-féle monoton majdnem teljes osztályokat és a következő eredményeket kaptuk:

1.) az $M$ határérték-logikában minden $r$ lineáris rendezéshez tartozó monoton függvényosztály mejden teljes;

2.) az $M$ határérték-logikának kontinuum sok majdnem teljes osztálya van, sőt már a lineáris rendezésre nézve monoton függvények majdnem teljes osztályainak a száma is kontinuum.

## 2. § Alapfogalmak

2.1. **Definició.** [4]

Legyen $E_k$ egy tetszőleges $k$-elemű halmaz. Jelölje $P_k^n$ $(n = 0,1,2,\dots)$ az olyan $n$ változós $f(x_1,\dots,x_n)$ függvények halmazát, amelyek változói és értékei az $E_k$ halmaz elemei.

$P_k = \bigcup\limits_{n=0}^{\infty} P_k^n$ függvényhalmazt $k$-értékü logikának nevezzük. Az általánosság megszoritása nélkül feltehetjük, hogy $E_k = \{0,1,\dots,k-1\}$, $k \geqslant 2$.

2.2. **Definició.**

Ha a 2.1. definicióban szereplő $E_k$ véges halmazt valamilyen $\aleph_0$ számosságú $E_{\aleph_0}$ halmazzal helyettesitjük, akkor az igy kapott $P_{\aleph_0}$ függvényhalmazt végtelenértékű logikának nevezzük.

$E_{\aleph_0}$ a továbbiakban mindig a nem negativ egész számok halmazát jelöli, azaz $E_{\aleph_0} = \{0,1,2,\dots,n,n+1,\dots\}$.

2.3. **Definició.** [6]

A $P_{\aleph_0}$ végtelenértékű logika $P$ részhalmazát határérték-logikának nevezzük, ha teljesülnek a következő feltételek:

a.) $P$ függvényosztályban csak megszámlálhatóan végtelen sok függvény van.

b.) minden természetes $k$ számhoz $(k \geqslant 2)$ létezik olyan $A_k$ függvényhalmaz $(A_k \subseteq P)$, amelyet homorf módon le lehet képezni a $k$-értékű logikára.

Legyen $P$ határérték-logika, $\epsilon$ az $E_{\aleph_0}$ részhalmaza $(\epsilon \subset E_{\aleph_0})$ és $g(x_1,\dots,x_n) \leftarrow P$.

2.4. **Definició.**

Jelölje $g_\epsilon(x_1,\dots,x_n)$ azt a függvényt, amely az $\epsilon$ halmaz $n$-szeres szorzathalmazán egyenlő a $g(x_1,\dots,x_n)$ függvénnyel és azonkivül 0.

A $g_\epsilon(x_1,\dots,x_n)$ függvényt a $g(x_1,\dots,x_n)$ függvény halmazra vonatkozó szükitésének nevezzük.

2.5. **Definició.**

Azt mondjuk, hogy a $A_k (A_k \subset P_{\aleph_0})$ függvényhalmaz a $k$-értékű logika modellje az $\epsilon_k = \{e_0, e_1, \dots, e_{k-1}\}$ $(k \geqslant 2)$ halmazon $(\epsilon_k \subset E_{\aleph_0})$, ha az $[A_k]$ halmazban létezik olyan $f(x_1,x_2)$ függvény, hogy

$$f_{\epsilon_k}(x_1,x_2) = \begin{cases} e_{i+1}, & \text{ha } (x_1,x_2) \leftarrow \epsilon_k x \epsilon_k \text{ és max } (x_1,x_2) = e_i, \\ & \text{ahol } 0 \leqslant i \leqslant k-2, \\ & \\ e_0, & \text{ha } (x_1,x_2) \leftarrow \epsilon_k x \epsilon_k \text{ és max } (x_1,x_2) = e_{k-1}, \end{cases}$$

$(e_0 < e_1 < \ldots < e_{k-1})$.

Könnyü belátni, hogy ha $A_k$ a $k$-érékű logika modellje $\epsilon_k$ halmazon, akkor az $A_k$ függvényhalmazban van legalább egy olyan függvényhalmaz, amelynek a szükitése az $\epsilon_k$ halmazra izomorf a $k$-értékü logikával.

2.6. **Definició.**[2]

A véges $\epsilon_k$ részhalmazok rendszerét az $A$ függvényhalmaz tartományának nevezzük és $T_A$-val jelöljük. Az $\epsilon_k = \{ e_0, e_1, \ldots, e_{k-1} \}$ halmaz akkor és csak akkor tartozik a $T_A$ tartományhoz, ha a $k$-értékű logika modellje az $\epsilon_k$ halmazon.

2.7. **Definició.** [2]

A $P$ határérték-logikát növekvőnek nevezzük, ha a $T_p$ tartománya tartalmaz olyan véges halmazok végtelen sorozatát $(\Pi = \{ \epsilon_2, \epsilon_3, \ldots, \epsilon_i, \ldots \})$, amelyre igaz, hogy

$\epsilon_i \subset \epsilon_{i+1}$ $(i = 2,3, \ldots)$.

2.1. **Tétel.** [2]

*A határérték-logika maximális akkor és csak akkor, ha növekvő. A szuperpozició, zártság a majdnem teljesség fogalmat hasonlóan értelmezzük, mint* [12]-ben.

### 3. § Az M határérték - logika

Határozzuk meg a maximális határérték-logika egy M reprezentását, amelyet a továbbiakban tanulmányozni fogunk:

Definiáljuk a $\mu_k(x_1,x_2) \leftarrow P_{\aleph_0}$ $(k \geqslant 2)$ függvényt a következőképpen:

$$\mu_k(x_1,x_2) = \begin{cases} e, & \text{ha } (x_1,x_2) \leftarrow \epsilon_k x \epsilon_k \text{ és } \max(x_1,x_2) = e-1, \\ & \text{ahol } 1 \leqslant e-1 \leqslant k-1 \\ 1, & \text{ha } (x_1,x_2) \leftarrow \epsilon_k x \epsilon_k \text{ és max } (x_1,x_2) = k, \\ 0 & \text{különben} \end{cases}$$

Jelölje $M_k$ a $\mu_k(x_1,x_2)$ függvényhalmaz lezártját

$([\{ \mu_k(x_1,x_2) \}])$ és $M = [\bigcup_{k=2}^{\infty} M_k]$.

3.1. **Megjegyzés.**

Könnyű belátni, hogy $M_k$ izomorf a $P_k$-val és hogy $M$ határérték-logika.

3.2. **Megjegyzés.**

Illusztrációként megadjuk a $\mu_2(x_1,x_2)$, $\mu_3(x_1,x_2)$ illetve $\mu_k(x_1,x_2)$ függvényeket

| $x_1/x_2$ | 0 | 1 | 2 | 3 | 4 ..... |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 2 | 1 | 0 | |
| 2 | 0 | 1 | 1 | 0 | 0 |
| 3 | 0 | 0 | 0 | 0 | |
| 4 | 0 | | 0 | | 0 |

$\epsilon_k = \{1,2\}$

$\mu_2(x_1,x_2)$

| $x_1/x_2$ | 0 | 1 | 2 | 3 | 4 ..... | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | |
| 1 | 0 | 2 | 3 | 1 | | |
| 2 | 0 | 3 | 3 | 1 | | 0 |
| 3 | 0 | 1 | 1 | 1 | | |
| 4 | 0 | | | | 0 | |
| | | | 0 | | | 0 |

$\epsilon_k = \{1,2,3\}$

$\mu_3(x_1,x_2)$

| $x_1/x_2$ | 0 | 1 | 2 | 3 | 4 .... | $k-1$ | $k$ | $k+1$ ... |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 .... | 0 | 0 | 0 ....... |
| 1 | 0 | 2 | 3 | 4 | 5 .... | $k$ | 1 | |
| 2 | 0 | 3 | 3 | 4 | 5 .... | $k$ | 1 | |
| 3 | 0 | 4 | 4 | 4 | 5 .... | $k$ | 1 | |
| 4 | 0 | 5 | 5 | 5 | 5 .... | $k$ | 1 | 0 |
| $k-1$ | 0 | $k$ | $k$ | $k$ | $k$ .... | $k$ | 1 | |
| $k$ | 0 | 1 | 1 | 1 | 1 .... | 1 | 1 | |
| $k+1$ | 0 | | | | 0 | | | |

$\epsilon_k = \{1,2,\ldots,k\}$

$\mu_k(x_1,x_2)$

3.1. **Tétel.**

*M hatarérték logika.*

**Bizonyitás.**

A 2.3 definició alapján be kell látnunk, hogy

a.) $M$ függvényosztályban csak megszámlálhatóan végtelen sok függvény van.

Mivel $M = [\bigcup_{k=2}^{\infty} M_k]$ ahol mindenegyes $M_k$ függvényosztály megszámlálhatóan végtelen sok függvényt tartalmaz és ilyen függvényosztályunk megszámlálhatóan végtelen sok van, akkor az uniójuk is és a lezártjuk is megszámlálhatóan végtelen számosságu.

b.) Minden természetes $k$-számhoz ($k \geqslant 2$) létezik egy olyan függvényhalmaz, amely homomorf módon le lehet képezni a $k$-értékű logikára. Az állitás közvetlenül következik abból, hogy a $\mu_k(x_1,x_2)$ függvénynek kölcsönösen egyértelmüen megfeleltethetjük a

$W_k(x_1,x_2)$ Webb függvényt [5].

A továbbiakban legyen $\epsilon_k \subset E_{\aleph_0} \setminus 0$ részhalmaza, amelyről az általánosság megszoritása nélkül feltehetjük, hogy $\epsilon_k = \{1,2,\dots,k\}$.

.3.3. **Megjegyzés.**

Az $M$ határérték-logika függvényei rendelkeznek azzal a tulajdonsággal, hogy $f(x_1,\dots,x_{i-1},0,x_{i+1},\dots,x_n) = 0$.

3.2. **Tétel.**

*M maximális határérték-logika.*

**Bizonyitás.**

Az $M$ határérték-logika maximalitása közvetlenül adódik abból, hogy : $M$ növekvő a 2.7 Definició értelmében és igy a 2.1.Tétel miatt ez szükséges és elegendő feltétel arra, hogy az $M$ határérték-logika maximális legyen.

## 4. §. Az M határérték-logika monoton függvényosztályai

Ebben a paragrafusban az M határérték-logika monoton osztályait vizsgáljuk. Bebizo nyitjuk, hogy az $M$ monoton osztályai a $P_k$ monoton osztályaihoz hasonlóak és számosságuk kontinuum. Sőt, az $M$ határérték-logikában kontinuum sok monoton majdnem teljes osztály van, mig a $k$-értékü logikában véges sok.

4.1. **Megjegyzés.**

Természetszerüen nem az $M$ függvényosztályról, hanem annak egy $\epsilon_\gamma$ halmazra való $M_{\epsilon_\gamma}$ megszoritásáról beszélünk (ez vonatkozik az $f(x_1,\dots,x_n) \leftarrow M$ függvényekre is)

Ez nem megszoritás a tételekre.

Az $\epsilon_\gamma$ halmazt egy $f(x_1, \dots, x_n) \leftarrow M$ függvényből kiindulva a következőképpen kapjuk:

1.) Jelölje $\alpha$ az $f(x_1, \dots, x_n)$ függvény által felvett értékek maximumát

2.) $\beta$ az a legnagyobb érték, amely valamely szám $n$-esben előfordulva még nem ad 0-t ($f(\dots, p, \dots) \neq 0$)

3.) $\gamma = \max(\alpha, \beta)$, $\epsilon_\gamma = \{1, 2, \dots, \gamma\}$, $(k \geqslant \gamma)$.

4.1. **Definició.**

$\tilde{\alpha} \leqslant \tilde{\beta}$, ha $\alpha_i, \leqslant \beta_i$, ahol $\alpha_i, \beta_i \leftarrow E_k (i = 1, \dots, n)$.

Azt mondjuk, hogy az $f(x_1, \dots, x_n) \leftarrow P_k (P_{\aleph_0})$ függvény monoton a lineáris rendezésre nézve, ha tetszőleges $\tilde{\alpha}, \tilde{\beta} \leftarrow E_k (E_{\aleph_0}) \tilde{\alpha} \leqslant \tilde{\beta}$ kombináció párra $f(\tilde{\alpha}) \leqslant f(\tilde{\beta})$.

4.2. **Megjegyzés.**

$<$ -val jelöljük az $1 < 2 < 3 < \dots < n < n+1$ rendezést. A $<_r$ lineáris rendezésben a 0 legyen egy kitüntetett elem, amely egyetlen $E_{\aleph_0} \setminus 0$ elemmel sem összehasonlitható. Jelöljük $M^r_{\epsilon_k}$ -val az $\epsilon_k$ feletti $r$ rendezéshez tartozó összes monoton függvények halmazát. Ekkor

$$M^r = \bigcup_{k=2}^{\infty} M^r_{\epsilon_k}.$$

Tekinteni fogunk egy tetszőleges lineáris rendezést és bebizonyitjuk, hogy ezen rendezésre nézve a monoton függvények majdnem teljesek és különböző lineáris rendezésekhez különböző majdnem teljes osztályok tartoznak.

4.3. **Megjegyzés.**

$M^r_{\epsilon_k}$ mindazon $f(x_1, \dots, x_n) <_r$ szerint monoton függvények halmaza az $M^r$ monoton függvényosztályból, amelyekre igaz, hogy $f(x_1, \dots, e, \dots, x_n) = 0$ ha $e \not\leftarrow \epsilon_k$.

4.1. **Tétel.**

*Az $M^r_{\epsilon_k}$ és $M^r$ függvényosztályok zártak.*

Bizonyitás.

Helyettesitsünk a változók helyébe $<_r$ szerint monoton függvényeket.

Legyen

$$f_{\epsilon_k}(x_1, x_2, \dots, x_n) = g_{\epsilon_k}(g_{\epsilon_k,1}(x_1, x_2, \dots, x_n), g_{\epsilon_k,2}(x_1, x_2, \dots, x_n), \dots,$$
$$, \dots, g_{\epsilon_k,m}(x_1, x_2, \dots, x_n))$$

ahol $g_{\epsilon_k}, g_{\epsilon_k,1}, g_{\epsilon_k,2}, \ldots, g_{\epsilon_k,m}$ függvények $<_r$ szerint monoton függvények. Megmutatjuk, hogy a $f$ függvény szintén monoton függvény az $<_r$ rendezésre nézve. Vegyünk két sorozatot $\tilde{\alpha}$-t és $\tilde{\beta}$-t ugy, hogy $\tilde{\alpha} <_r \tilde{\beta}$. Világos, hogy a feltétel miatt

$$g_{\epsilon_k,i}(\tilde{\alpha}) \leqslant_r g_{\epsilon_k,i}(\tilde{\beta}), \quad (i = 1,2, \ldots, m)$$

ezért a

$$\{ g_{\epsilon_k,1}(\tilde{\alpha}), \ldots, g_{\epsilon_k,2}(\tilde{\alpha}), \ldots, g_{\epsilon_k,m}(\tilde{\alpha}) \} \text{ és a } \{ g_{\epsilon_k,1}(\tilde{\beta}), g_{\epsilon_k,2}(\tilde{\beta}), \ldots, \ldots, g_{\epsilon_k,m}(\tilde{\beta}) \}$$

sorozatok olyanok, hogy

$$\{ g_{\epsilon_k,1}(\tilde{\alpha}) g_{\epsilon_k,2}(\tilde{\alpha}), \ldots, g_{\epsilon_k,m}(\tilde{\alpha}) \} \leqslant_r \{ g_{\epsilon_k,1}(\tilde{\beta}) g_{\epsilon_k,2}(\tilde{\beta}), \ldots, g_{\epsilon_k,m}(\tilde{\beta}) \}$$

Innen a $g_{\epsilon_k}$ függvény monotonitása miatt kapjuk, hogy

$$f_{\epsilon_k}(\tilde{\alpha}) = g_{\epsilon_k}(g_{\epsilon_k,1}(\tilde{\alpha}), \ldots, g_{\epsilon_k,m}(\tilde{\alpha})) \leqslant_r g_{\epsilon_k}(g_{\epsilon_k,1}(\tilde{\beta}), \ldots, g_{\epsilon_k,m}(\tilde{\beta})) = f_{\epsilon_k}(\tilde{\beta}).$$

Igy a vizsgált függvényosztály invariáns a szuperpozicióval szemben, tehát zárt osztály. A tétel be van bizonyitva.

Jelöljük $C^j_{\epsilon_k}(x)$-vel a következő függvényt:

| $x$ | 0 | 1 | . . . . . . . . . . . . . . . . . . . $k$ | $k+1$ . . . . . . . . . . . |
|---|---|---|---|---|
| $C^j_{\epsilon_k}(x)$ | 0 | $j$ | . . . . . . . . . . . . . . . . . . . $j$ | 0 . . . . . . . . . . . . . . . |

4.4. **Megjegyzés.**

Ha $<_r \equiv <$ rendezéssel, azaz azonos a szokásos $1 < 2 < 3 \ldots$ lineáris rendezéssel, akkor az e szerint a rendezés szerinti ismert monoton függvények $M^1$ osztályát kapjuk.

A következő néhány tétel és lemma bizonyitásának gondolatmenete Sz. V. Jablonszkij [12]-ben a $k$-értékü logika megfelelő tételeinek a bizonyitása alapján történt.

4.2. **Tétel.**

*Az* $M^1_{\epsilon_k}$ *monoton függvények osztályát a* $\max_{\epsilon_k}(x_1, x_2)$ $\min_{\epsilon_k}(x_1, x_2)$, $C^j_{\epsilon_k}(x)$ $(j = 1,2, \ldots, k)$ *és az* $m^i_{\epsilon_k}(x)$ $(i = 2, \ldots, k)$ *függvények generálják (amelyek* $M^1_{\epsilon_k}$ *-beliek), ahol*

$$m^i_{\epsilon_k}(x) = \begin{cases} 1, \ \textit{ha} & 1 \leqslant x > i \ \textit{és} \ \ x \leftarrow \epsilon_k; \\ k, \ \textit{ha} & k \leqslant x \geqslant i \ \textit{és} \ \ x \leftarrow \epsilon_k; \\ 0 \ \textit{különben}. \end{cases}$$

Bizonyitás.

Könnyű belátni, hogy az előbb felsorolt függvények monotonok.

Definiáljuk a $z^{\widetilde{\alpha},\beta}_{\epsilon_k}(x_1, \ldots, x_n)$ függvényt a következőképpen:

$$z^{\widetilde{\alpha},\beta}_{\epsilon_k}(x_1 \ldots, x_n) = \begin{cases} \beta, \ \text{ha} \ \widetilde{x} > \widetilde{\alpha} \ \text{és} \ x_i \leftarrow \epsilon_k (i = 1,2, \ldots, n); \\ 1, \ \text{ha} \ \widetilde{x} \not> \widetilde{\alpha} \ \text{és} \ x_i \leftarrow \epsilon_k (i = 1,2, \ldots, n); \\ 0 \ \text{különben}, \end{cases}$$

ahol $\widetilde{\alpha} = (\alpha_1, \ldots, \alpha_n)$ $(\alpha_i \leftarrow \epsilon_k)$.

Nyilvánvaló, hogy

$$z^{\widetilde{\alpha},\beta}_{\epsilon_k}(x_1, \ldots, x_n) = \min_{\epsilon_k}(\beta, m^{\alpha_1}_{\epsilon_k}(x_1), \ldots, m^{\alpha_n}_{\epsilon_k}(x_n)).$$

Legyen $z_{\epsilon_k}(x_1, \ldots, x_n)$ egy tetszőleges $M^1_{\epsilon_k}$-beli monoton függvény, akkor fennáll a következő egyenlőség:

$$z_{\epsilon_k}(x_1, \ldots, x_n) = \max_{\widetilde{\alpha} \ \epsilon_k} \{ z^{\widetilde{\alpha}, f(\widetilde{\alpha})}_{\epsilon_k}(x_1, \ldots, x_n) \}.$$

Ezzel a tétel be van bizonyitva.

4.1. **Lemma.**

Ha $z(x_1, \ldots, x_n) \leftarrow M_k$ nem monoton függvény, akkor $c^j$-k behelyettesitésének segitségével megkaphatjuk az egyváltozós nem monoton függvényt.

Bizonyitás

Elegendő, ha csak $\epsilon_k$-beli elemeket tekintünk és igy a bizonyitás teljesen azonosan történik mint [12]-ben

4.2. **Definició.**

A $B$ halmaz a $B_1, B_2, \ldots, B_e$ halmazok direk összege, azaz $B = B_1 + B_2 + \ldots + B_\ell$ ha

1.) $B = B_1 \cup B_2 \cup \ldots \cup B_e$;

2.) $B_i$-k páronként idegenek $(i = 1,2, \ldots, l)$ azaz

$$B_i \cap B_j = \emptyset, \quad \text{ha } i \neq j$$

4.1. 1. **Lemma** [12] .

*Ha a $B^*$ halmaz felbontható két $C$ és $D$ nem üres halmaz direkt összegére és $\tilde{\gamma}^0 \in C$, $\tilde{\delta} \in D$, akkor létezik két olyan elem $\tilde{\gamma} \in C$ és $\tilde{\delta} \in D$, hogy az egyik közvetlenül következik a másik után. Emellett, ha $\tilde{\gamma}^0 <_r \tilde{\delta}^0$, (vagy forditva $\tilde{\gamma}^0 >_r \tilde{\delta}^0$), akkor $\tilde{\gamma}^0 \leqslant_r \tilde{\gamma} <_r \tilde{\delta} \leqslant_r \tilde{\delta}^0$, (vagy forditva $\tilde{\gamma}^0 \geqslant_r \tilde{\gamma} >_r \tilde{\delta} \geqslant_r \tilde{\delta}^0$).*

Bizonyitás.

Lásd [12].

4.3. **Tétel.**

*Az $M^1$ monoton függvényosztály majdnem teljes az $M$ határérték-logikában, ahhol $M^1$ tetszőleges lineáris rendezés szerint monoton függvények osztálya.*

Bizonyitás

Azt kell bizonyitani, hogy $[\{ f(x_1, \ldots, x_n) \} \cup M^1] = M$. $f(x_1, \ldots, x_n) \in M^1$.

Legyen $k \geqslant \gamma$, ahol $\gamma = \max(\alpha, \beta)$; $\alpha = \max f(\tilde{x})$, $\beta$ maximuma azon értékeknek, amelyekre $f(\ldots, \beta, \ldots) \neq 0$. Mutassuk meg, hogy az $f_{\epsilon_k}(x_1, \ldots, x_n)$ függvény és az $M^1_{\epsilon_k}$ halmaz generálja a $[\{ \mu_k(x_1, x_2) \}] = M_k$ logikát, amely izomorf a $P_k$ $k$-értékü logikával.

Tehát legyen

$$f_{\epsilon_k}(x_1, \ldots, x_n) \not\leftarrow M^1_{\epsilon_k}.$$

Akkor a 4.1 Lemma szerint a $C^j_{\epsilon_k}$ függvények segitségével felépithetjük a $g_{\epsilon_k}(x)$ egy változótól függő nem monoton függvényt.

Tegyük fel, hogy $x = t$-re

$$g_{\epsilon_k}(t) > g_{\epsilon_k}(t+1) \quad (g(0) = 0).$$

$$k^1_{\epsilon_k}(x) = \begin{cases} t, & \text{ha } x = 1; \\ t+1, \text{ ha } x \neq 1 & \text{és } x \leftarrow \epsilon_k; \\ 0 \text{ különben}. & \end{cases}$$

$$k^2_{\epsilon_k}(x) = \begin{cases} 1, \text{ ha } x \leqslant g(t+1) & \text{és } x \leftarrow \epsilon_k; \\ k, \text{ ha } x > g(t+1) & \text{és } x \leftarrow \epsilon_k; \\ 0, \text{ ha } x \not\leftarrow \epsilon_k & \end{cases}$$

Könnyű belátni, hogy $k^1_{\epsilon_k}(x)$ és $k^2_{\epsilon_k}(x)$ monoton függvények, azaz $k^1_{\epsilon_k}(x)$, $k^2_{\epsilon_k}(x) \leftarrow M^1_{\epsilon_k}$ és

$$k^2_{\epsilon_k}(g_{\epsilon_k}(k^1_{\epsilon_k}(x)) = \begin{cases} k, & \text{ha } x = 1 \\ 1, & \text{ha } x \neq 1 \\ 0 & \text{különben.} \end{cases} \qquad \text{és } x \leftarrow \epsilon_k;$$

Igy $k^2_{\epsilon_k}(g_{\epsilon_k}(k^1_{\epsilon_k}(x))) \equiv j^1_{\epsilon_k}(x)$, ahol

$$j^i_{\epsilon_k}(x) = \begin{cases} k, & \text{ha } x = i \neq 0 \text{ és } x \leftarrow \epsilon_k; \\ 1, & \text{ha } x \neq i \quad\; \text{ és } x \leftarrow \epsilon_k; \\ 0 & \text{különben.} \end{cases}$$

Nézzük meg a következő függvényeket:

$$\phi^i_{\epsilon_k}(x) = \begin{cases} 1, & \text{ha } x < i \text{ és } x \leftarrow \epsilon_k; \\ k, & \text{ha } x \geqslant i \text{ és } x \leftarrow k; \\ 0 & \text{különben.} \end{cases}$$

$$\psi^i_{\epsilon_k}(x) = \begin{cases} 1, & \text{ha } x \leqslant i \text{ és } x \leftarrow \epsilon_k; \\ 2, & \text{ha } x > i \text{ és } x \leftarrow \epsilon_k; \\ 0 & \text{különben.} \end{cases}$$

Világos, hogy $\phi^i_{\epsilon_k}(x)$, $\psi^i_{\epsilon_k}(x) \leftarrow M^1_{\epsilon_k}$,

Következésképpen nekünk a következő függvényeink vannak:

$C^j_{\epsilon_k}(x)$, $\max_{\epsilon_k}(x_1, x_2)$, $\min_{\epsilon_k}(x_1 x_2)$ és $j^i_{\epsilon_k}(x)$ $(i = 1, 2, \ldots, k)$.

4.3.1 **Lemma.**

A $C^j_{\epsilon_k}(x)$, a $\max_{\epsilon_k}(x_1, x_2)$ és a $j^i_{\epsilon_k}(x)$ $(1 \leqslant i \leqslant k)$ teljesek $M_k$-ban $([\{\mu_k(x_1, x_2)\}] = M_k)$.

ahol

$$C^j_{\epsilon_k}(x) = \begin{cases} j, & \text{ha } x \leftarrow \epsilon_k; \\ 0 & \text{különben.} \end{cases}$$

$$j^i_{\epsilon_k}(x) = \begin{cases} k, & \text{ha } x = i \neq 0 \text{ és } x \leftarrow \epsilon_k; \\ 1, & \text{ha } x \neq i \text{ és } x \leftarrow \epsilon_k; \\ 0 & \text{különben.} \end{cases}$$

**Bizonyitás.**

A változók száma szerinti teljes indukcióval adjuk meg, ugy hogy az adott függvényrendszerekből megkaphatjuk tetszőleges $M_k$-beli függvényt.

Legyen $f(x_1, \ldots, x_n)$ egy tetszőleges függvény az $M_k$–ból.

1.) Ha $f$ nem függ változótól, azaz a $C^j_{\epsilon_k}(x)$ $(j = 1,2, \ldots, k)$ függvény, akkor az állitás triviális, mivel a kiinduló rendszer tartalmazza $j$-ket $(j = 1,2, \ldots, k)$.

2.) Tegyük fel, hogy kiindulva a függvényrendszerből a szuperpozició segitségével, fel lehet épiteni az összes $M^1_{\epsilon_k}$-beli $n$-változós függvényt. Megmutatjuk, hogy elő tudunk állitani egy tetszőleges $n+1$ változóst függvényt $M^1_{\epsilon_k}$-ból és a 4.3.1 Lemmában adott függvényekből. Ezért megjegyezzük, hogy ha feltesszük

$$\max_{\epsilon_k}(y_1, y_2, \ldots, y_n) = \max_{\epsilon_k}\{\max_{\epsilon_k}[\ldots \max_{\epsilon_k}(\max_{\epsilon_k}(y_1, \ldots, y_n)\ldots]y_n\}$$

(hasonlóan $\min_{\epsilon_k}(y_1, y_2, \ldots, y_n)$-re),

akkor

$$f_{\epsilon_k}(x_1, \ldots, x_n, x_{n+1}) = \max_{\epsilon_k}\{\min_{\epsilon_k}[j^1_{\epsilon_k}(x_{n+1}), f(x_1, \ldots, x_{n-1}\, 1)]$$
$$\min_{\epsilon_k}[j^2_{\epsilon_k}(x_{n+1}), f(x_1, \ldots, x_n\; 2)] \min_{\epsilon_k}[j^k_{\epsilon_k}(x_{n+1}), f(x_1, \ldots, x_n\; k)]\}$$
$$(\epsilon_k = 1,2, \ldots, k).$$

Innen azonnal kapjuk az állitást.

A fenti lemma értelmében tehát ez a rendszer teljes.

Igy a tétel be van bizonyitva.

4.3 Tételből következik:

**4.4. Tétel.**

*Az M határérték-logikában minden lineáris rendezéshez tartozó monoton függvényosztály ($M^r$) majdnem teljes.*

Bizonyitás

Amennyiben a $M^r_{\epsilon_k}$ függvényhalmaz duálisa az $M^1_{\epsilon_k}$ függvényhalmaznak

$$s_{\epsilon_k}(x) = \begin{pmatrix} a_1, a_2, \ldots, a_k \\ \\ 1, 2, \ldots, k \end{pmatrix}$$

(ahol $\epsilon_k$ halmaz elemei a következőképpen van elrendezve $a_1 <_r a_2 <_r \ldots <_r a_k$)

előállitására vonatkozóan, akkor az itt $M^1_{\epsilon_k}$ osztályra meghatározott eredményeket átvihetjük (a dualitás elve szerint) az $M^r_{\epsilon_k}$ függvényosztályra.

4.1. **Következmény.**

Minden $M^r_{\epsilon_k}$ függvényosztály majdnem teljes $M_k$-ban.

4.5. **Tétel.**

*Az $M$ határérték-logikában majdnem teljes monoton osztályok száma kontinuum.*

**Bizonyitás.**

Mivel az $M$ megszámlálható sok függvényből áll, igy több mint kontinuum sok majdnem teljes osztály az $M$-ben nem is lehet.

Az, hogy az $M^r$ osztályok száma kontinuum következik abból, hogy

a.) a különböző $<_r$ rendezések száma kontinuum,

b.) a különböző rendezésekhez különböző $M^r$-ek felelnek meg.

4.2. **Következmény.**

Az $M$ határérték-logikában kontinuum sok majdnem teljes osztály van.

## I r o d a l o m


[1] J. Demetrovics: A határérték-logikák homomorfizmusairól Alk. Mat. 1. (1975.) (125-138).

[2] J. Demetrovics: Az M maximális határérték-logikáról Alk. Mat. 2 (1976), (57-66).

[3] S.L.Lec, E.T. Lee: On Multivalued Symmetric Functions, IEEE. Trans. on Computers C-21. (1972.) (312-317)

[4] I. Rosenberg: Uber die Verschi den keit maximaler Klassen Rev. Roum. math jures et appl. 14, (1969) (413-438)

[5] D.L. Webb: Generation of any n-valued logic by one binary operator. Proc. Mat. Acad Sci 21. (1935) (252-254).

[6] Г.П. Гаврилов: О мощности множества предельных логик, обладающих конечным базисом, Сб. "Проблемы кибернетики" вып. 21, М., "Наука" /1969/, /113-126/.

[7] В.М. Гниденко: Нахождение порядков предельных классов в трехзначной логике, Сб. "Проблемы кибернетики", вып. 8, М., /1962/, /341-346/.

[8] Я. Деметрович: О числе попарно неизоморфных предельных логик, Сб. "Дискретный анализ", вып. 24, Новосибирск, "Наука", /1974/, /21-29/.

[9] Я. Деметрович: О сравнении предельных логик при моделировании в них конечнозначных логик, Акта Кибернетика, /1974/, /2/.

[10] Я. Деметрович: О свойствах минимальной предельной логики, Штудиа Мат. Акад. Шси. Нунг. 9., /1974/, /1-2/.

[11] Я. Деметрович: О некоторых гомоморфизмах и отношениях для предельных логик, Сб. "Проблемы кибернетики" 30., Москва, /1975/, /5-42/.

[12] С.В. Яблонский: Функциональные построения в к-значной логике, Труды Мат. АН СССР, 51., /1958/, /5-142/.

[13] С.В. Яблонский: О предельных логиках, Докл. АН СССР 118.4. /1958/, /657-660/.

[14] С.В. Яблонский: О некоторых свойствах счетных замкнутых классов из $P_{\aleph_0}$ , Докл. АН СССР 124.5., /1959/, /990-993/.

[15] С.В. Яблонский, Г.П. Гаврилов, В.Б. Кудрявцев: Функции алгебры логики и классы Поста, "Наука", Москва, /1966/.

## Р Е З Ю М Е

### О монотонных классах предельной логики M

Ева Гардош

В настоящей работе мы докажем, что в предельной логике M имеется континуум монотонных предполных классов. Нам удалось обобщить линейные монотонные предполные классы типа С.В. Яблонской и получили следующие результаты:

- в предельной логике M каждому линейному упорядочению соответствует монотонный предполный класс;
- для каждого линейного упорядочения множества $E_{\aleph_0}$ имеется континуум монотонных предполных классов.



## S u m m a r y

### On the monotone classes of the M limit-logic

Eva Gardos

In this paper we prove, that in the limit-logic which is examined by us, containes continuum monotone almost-complete classes.

We succeed in generalizing the monotone almost-complete classes defined by Jablonszkij and we obtained the next results:

1.) In the M limit-logic any monotone function-classes belong to every r linear order are almost-complet classes, moreover the number of the almost-complete classes of the monotone functions look at linear order is already continuum.

MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények** 20/1978.

# RELÁCIÓS ADATBÁZIS MODELL

Demetrovics János

## 1. Bevezetés

Az adatbázis a valóságról adatokat tartalmaz. Az adatok a valóság objektumainak tulajdonságaira vonatkozhatnak, s ezek közül a tulajdonságok közül a kapcsolatot jelentő tulajdonság az adatbázisban sok szemléleti problémát okoz, egészen az "adatok" és "kapcsolatok" megkülönböztetéséig. A probléma kialakulása a fejlődés következménye: a homogén rekordokból álló file-ok a valóság objektumait a rekordoknak feleltették meg, kapcsolat ábrázolására minimális információt tartalmaznak. A közvetlen hozzáférésű tárak adták meg a valóság objektumai közötti kapcsolatok ábrázolásának lehetőségeit, rekordok közötti kapcsolatok formájában. Igy alakultak ki a kapcsoló adatok, s a valóság objektumai közötti kapcsolat igy vált rekordok közötti kapcsolattá. A következőkben ez a kettősség nem lesz zavaró, ha leszögezzük, hogy a kapcsolat ábrázolására az adatbázisban több lehetőségünk is lehet, s az adatbázison belüli kapcsolaton mindig az adatbázis objektumai közötti kapcsolatot értünk.

Az adatbáziskezelő rendszereket többféleképpen szoktuk osztályozni, de a leginkább használatos osztályozás éppen a szerint történik, hogy a felhasználó szempontjából miként valósul meg a kapcsolatokkal való manipulálás. E szerint három adatbázis rendszer tipust különböztetünk meg: hierarchikust, hálózatost és relációst.

A relációs megközelitésben a kapcsolatokat ugyanugy ábrázoljuk, mint a valós világ többi adatát, azaz $n$-esek segitségével. Ez azt jelenti, hogy a relációs adatmodell a valóságot és a kapcsolatot ugyanazon tipusu objektumnak tekinti. A relációs adatbázis modell nem a gépi oldal lehetőségeit igyekszik kihasználni, illetve bőviteni, hanem a felhasználó szempontjából indul ki és olyan eszközöket igyekszik kialakitani, amelyek segitségével ugy kerülünk a számitógéphez közelebb, hogy nincs szükség ujabb, magasabb szintü ismeretek elsajátitására. A hagyományos adatbázis modell a gépi hatékonyságot tekinti elsődlegesnek, mig a relációs adatmodell az adatfüggetlenséget még inkább kihangsulyozza és fontosnak tartja.

A hagyományos adatbázis modell a gépi hatékonyság lehetőségeit általában ugy tudja növelni, hogy az emberi oldalon valamilyen módon növeli a bonyolultságot. Ez álatában a felhasználók körének szükülését vonja maga után.

A relációs adatmodell a gép hatékonyságát másodlagosnak tekinti és ezért az adatokat a lehető legtermészetesebb formában, mátrix alakban szemléli.

Dolgozatunkban a relációs adatmodellről kivánunk egy áttekintést nyujtani.

## 2. Relációs adatbázis modell

A relációs adatbázis modell elmélete időben legkésőbb fejlődött ki. 1968-ban Childs [1] vetette fel a relációs elmélet adatbázis modellezésben történő felhasználását, amelyet

E.F. Codd [2,3,4] a későbbi években jelentős részben meg is valósitott.

A relációs adatbázis modell a relációk matematikai elméletén alapszik. Ez komoly elméleti alapot ad az adatbázis modell számára és a relációs elmélet számos eredményét alkalmazni lehet olyan fontos problémák megoldására, mint például az adatbázis kezelő /al/nyelv [5,6,7] megkonstruálása. Másrészről azonban a felhasználónak szembe kell nézni azzal a ténnyel, hogy uj kifejezéseket is meg kell tanulni a már eddig is használt hasonló fogalmakra.

A relációs adatbázis azon az alapvető törekvésen alapszik, hogy a felhasználó számára az adatokat a lehető legszemléletesebb módon tárgyalja. Nyilvánvaló, hogy a legszemléletesebb adatszerkezet a táblázat, vagyis egy olyan mátrix amelynek $m$ sora és $n$ oszlopa van $(m \times n)$. Lényegében a relációs adatbázis modell $t$ darab két dimenziós mátrixból áll: $m_i \times n_i$, $1 \leqslant i \leqslant t$.

Matematikailag a relációt a következőképpen értelmezzük:

**Definició.** Legyenek adva $D_1, D_2, \dots, D_n$ halmazok, amelyek nem feltétlenül különbözőek. Azt mondjuk, hogy $R$ egy reláció ezen $n$ halmaz $D_1 \times D_2 \times \dots \times D_n$ direkt szorzatán, ha létezik, olyan $g(x_1, x_2, \dots, x_n)$ predikát, melynek segitségével minden szám $n$-esről $(a_1, a_2, \dots, a_n)$ el tudjuk dönteni, hogy $(a_1, a_2, \dots, a_n) \in R$ vagy pedig $(a_1, a_2, \dots, a_n) \bar{\in} R$, ahol $a_i \in D_i$, $1 \leqslant i \leqslant n$.

Más szóval az $R = \{ (a_1, a_2, \dots, a_n) : a_i \in D_i$ $(i = 1, 2, \dots, n)$ és $g(a_1, a_2, \dots, a_n) = 1$ halmazt relációnak nevezzük. $D_i$ a reláció $i$-edik értelmezési tartománya, és az értékkészlet nem más mint a $D_i$ halmazban lévő elemek. $n$-t az $R$ reláció fokának, $m$-t pedig az $R$ reláció számosságának nevezzük, ahol $m = |R|$ és $m$ a relációban lévő $n$-esek számát jelenti.

Az igy definiált $R$ relációt részletesebben jelöljük a következőképpen $R(D_1, D_2, \dots, D_n$ Vagyis $R$ a reláció neve, a zárójelben pedig az értelmezési tartományok nevei vannak feltüntetve. Ha az értelmezési tartományok közül bizonyosakat ki kivánunk tüntetni, akkor azokat aláhuzzuk.

DIÁK (<u>AZONOSITÓ</u>, NÉV OSZTÁLY)

| AZONOSITÓ | NÉV | OSZTÁLY |
|---|---|---|
| D1 | FEKETE | 3 |
| D2 | KISS | 3 |
| D3 | KISS | 4 |
| D4 | KESERÜ | 5 |

1. ábra

Az 1. ábra a DIÁK relációt mutatja, amely az AZONOSITÓ, NÉV, és az OSZTÁLY tartományokon van értelmezve. Az OSZTÁLY tartomány például a 3,4 és 5 értéket tartalmazza vagyis azt, hogy a diákok melyik osztályba járnak. A DIÁK reláció számossága 4, foka pedig 3.

Amint az 1. ábra is mutatja, a relációt legegyszerübben táblázatok segitségével tudjuk ábrázolni, ahol mindegyik sor $n$ adatot tartalmaz. Vegyük észre, hogy az adatsor illetve adatoszlop is táblázat.

A táblázat oszlopai az adott egyed tipusra jellemző tulajdonságokat tartalmazzák, a sorai pedig konkrét előfordulásokat határoznak meg, az érték készlet konkrét értékeivel. Vagyis a két dimenziós mátrix esetén egy sor és egy oszlop találkozásában egyetlen érték található, amely egy konkrét tulajdonság konkrét értékét adja meg. Meg kell jegyezni, hogy ez az érték nulla vagy üres elem is lehet.

**Definició.** Az olyan értelmezési tartományt vagy értelmezési tartományok halmazát, melyek értékei egyértelmüen azonositják a relációt, a reláció elsődleges kulcsának nevezzük.

Az 1. ábrán bemutatott DIÁK relációból észrevehetjük a relációk főbb tulajdonságait.

A táblázat tanulmányozásánál vegyük észre a következő tulajdonságokat:

1.) Nincs két azonos sor.
Ez a tulajdonság azt a természetes követelményt takarja, hogy két különböző egyed előfordulása legalább egy adott tulajdonságban el kell, hogy térjen egymástól.

2.) A sorok sorrendje lényegtelen.
A sorokra egy vagy több azonos értelmezési tartomány konkrét értékével hivatkozunk ami valójában nem más mint az elsődleges kulcs. Az 1.) tulajdonságból következik, hogy elsődleges kulcs mindig létezik, legrosszabb esetben az egész sor az elsődleges kulcs.

A továbbiakban az elsődleges kulcs értelmezési tartományait alá fogják huzni.

Az 1. ábrán lévő DIÁK relációban minden diák kap egy azonositót, amely egyértelmüen azonositja a diákot. Vegyük észre, hogy a NÉV és OSZTÁLY értelmezési tartományok nem feltétlenül azonositják a DIÁK reláció konkrét sorait. Ugyanez az eset áll fenn a 2. ábrán megadott TANÁR reláció esetében is. A TANÁR-DIÁK reláció esetén (3.ábra) két értelmezési tartomány van az AZONOSITÓ-T és az AZONOSITÓ-D.

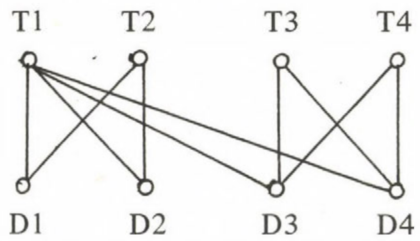
TANÁR (AZONOSITÓ, NÉV, TANTÁRGY)

| AZONOSITÓ | NÉV | TANTÁRGY |
|---|---|---|
| T1 | SOLYOM | ANGOL |
| T2 | NAGY | FIZIKA |
| T3 | FEHÉR | FIZIKA |
| T4 | NAGY | MAGYAR |

2. ábra

TANÁR–DIÁK (AZONOSITÓ-T, AZONOSITÓ-D)

| AZONOSITÓ-T | AZONOSITÓ-D |
|---|---|
| T1 | D1 |
| T1 | D2 |
| T1 | D3 |
| T1 | D4 |
| T2 | D1 |
| T2 | D2 |
| T3 | D3 |
| T3 | D4 |
| T4 | D3 |
| T4 | D4 |




3. ábra

Ezek a tartományok a TANÁR, illetve a DIÁK relációban valóban azonositottak egy-egy tanárt, illetve diákot, azonban a TANÁR-DIÁK relációban csak az együttes értékük (vagyis a teljes sor) azonosit egy-egy sort, amely értelemszerüen nem más, mint az, hogy ki kit tanit. Vegyük észre, hogy a TANÁR-DIÁK reláció segitségével valósitjuk meg azt az $M : N$ kapcsolatot, ami a tanárok és diákok között van. Vagyis azt, hogy egy tanár sok diákot tanit és egy diák több tanártól tanul.

**Definició.** Ha az elsődleges kulcs több értelmezési tartományból áll, akkor összetett kulcsról beszélünk.

3. Az oszlopok sorrendje lényegtelen

Az oszlopokra mindig az értelmezési tartomány nevével hivatkozunk és sohasem az oszlop poziciójával. Minden sorban az azonos tulajdonságok azonos helyen szerepelnek.

Nehézség merül fel abban az esetben, ha egy $R$ relációban ugyanaz az értelmezési tartomány többször szerepel. Ilyenkor ezeknek az értelmezési tartományoknak a neveit megkülönböztetjük valami jellel. Például a TANÁR-DIÁK relációban a TANÁR és a DIÁK relációból kiindulva értelemszerüen kétszer kellene, hogy az AZONOSITÓ értelmezési tartomány szerepeljen. A bonyodalmak elkerülése végett a TANÁR reláció AZONOSITÓ-ját a TANÁR-DIÁK relációban AZONOSITÓ-T-nek hivjuk, mig DIÁK reláció AZONOSITÓ-ját a TANÁR-DIÁK relációban AZONOSITÓ-D-nek. Vegyük észre, hogy az AZONOSITÓ-D-t jelölhettük volna egyszerüen AZONOSITÓ névvel is.

**Megjegyzés.** Codd müveiben, s több más dolgozatban is az oszlopok sorrendje lényeges. Ezekben a dolgozatokban kihasználják a reláció oszlopainak a pozicióját.

## 3. Funkcionális függőség

Amikor egy $R$ relációt jelentés szerint vizsgálunk, akkor bizonyos értelmezési tartományok közt függőséget tapasztalhatunk.

Ezek közül hármat kivánunk részletesebben megvizsgálni a funkcionális függőségét, a teljes funkcionális függőséget és a tranzitiv függőségét.

**Definició.** Egy adott $R$ reláció $Y$ értelmezési tartománya funkcionálisan függ az $X$ értelmezési tartománytól, akkor és csak akkor, ha minden $R$-beli $Y$ értéket meg határoz. Ezt lehet formálisan definiálni;

Ha $R = R(\ldots, X, \ldots, Y, \ldots)$ és $\tilde{a}_1 \epsilon R$, $\tilde{a}_2 \epsilon R$, ahol $\tilde{a}_1 = (a_{11}, a_{12}, \ldots, a_{1n})$ $\tilde{a}_2 = (a_{21}, a_{22}, \ldots, a_{2n})$, akkor létezik olyan $Y = f(X)$ függvény, hogy $f(a_{ix}) = a_{iY}$ $(i = 1,2)$.

Másszóval ez azt jelenti, hogy az $Y$ tulajdonság értékét az $X$ tulajdonság értéke adja meg.

**Megjegyzés.** A funkcionális függőség vonatkozhat összetett tartományra is, azaz $X = (X_1, X_2, \ldots, X_m)$, ahol $R = R(\ldots, X_1, \ldots, X_m, \ldots, Y, \ldots)$.

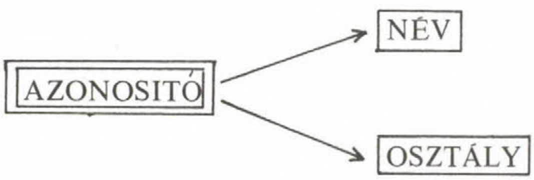
**Definició.** Legyen $X$ összetett tartomány. Teljes függőségéről beszélünk, ha az $Y$ értelmezési tartomány funkcionálisan függ az $X$ tartománytól, de nem függ az $X$ semmilyen valódi részhalmazától sem. Továbbiakban funkcionális függőség alatt mindig funkcionális teljes függőséget értünk, hacsak nem hangsulyozzuk ki, hogy funkcionális nem teljes függőségre gondoltunk.

**Megjegyzés.** Nyilvánvaló, hogy az elsődleges kulcstól minden más értelmezési tartomány funkcionálisan – esetleg teljes módon – függ. Amennyiben a kulcs nem összetett úgy a funkcionális függőség mindig funkcionális teljes függőséget jelent. Bizonyos relációkban léteznek olyan tartományok, amelyek az adott relációban nem elsődleges kulcsok, de egy másik relációban ezek a tartományok egyértelmüen azonosithatnak bizonyos tartományokat. Ezeket a tulajdonságokat reprezentáló értékeket idegen kulcsoknak nevezzük.

**Definició.** Az $R$ reláció $Z$ értelmezési tartománya tranzitive függ az $X$-től, ha $Z$ funkcionálisan függ az $X$ elsődleges kulcstól, és függ az $Y$ idegen kulcstól is, ami természetszerüen függ az $X$ elsődleges kulcstól.

Illusztráljuk példánkon keresztül ezeket a függőségeket.

A DIÁK relációban a NÉV és OSZTÁLY tartományok funkciónálisan függnek a DIÁK reláció AZONOSITÓ tartományától.




4. ábra

Ezeket a funkcionális függőségeket diagrammal ábrázolhatjuk, amint azt a 4. ábra mutatja.

A funkcionális függések felismerése az egyik legfontosabb és legkényesebb része az adatok szemantikájának. Csak a függések teljes feltárása után mondhatjuk azt, hogy az adatokat teljesen megértettük.

### 4. A relációk normalizálása

A relációk normalizálását illetve a normálformáinak fogalmát Codd vezette be, ő határozta meg az 1., 2. és a 3. normálforma tulajdonságait is [2, 5, 6].

Nézzük az MTA INTÉZET relációt ( 5. ábra)

MTA INTÉZET (M#, NÉV, TELEPHELY, CIM)

| M | NÉV | TELEPHELY | CIM |
|---|---|---|---|
| M1 | MTA SzTAKI | FŐ ÉPÜLET<br>VÁR<br>SZÁMITÓKÖZPONT | KENDE<br>URI<br>VICTOR HUGO |
| M2 | MTA KFKI | FŐ ÉPÜLET<br>CSILLAGVIZSGÁLÓ | KONKOLY<br>KONKOLY |

5. ábra

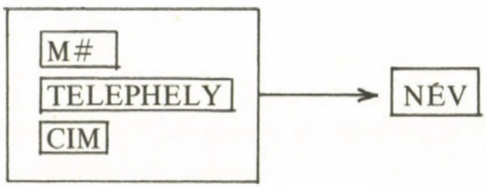
Rendes mátrix formában ezeket az adatokat nehezen tudjuk ábrázolni. A nehézség abból az egyszerü tényből adódik, hogy az értelmezési tartományban lévő értékek nem elemiek, hanem tovább lehet bontani kisebb elemekre.

**Definició.** Az *R* relációt 1 normálformájunak nevezzük, ha a relációban szereplő minden érték elemi.

Könnyü belátni, hogy minden relációt 1 normálformára lehet hozni. Ennek érdekében minden ismétlődő elemnél meg kell ismételnünk az egyeszeres elemeket.

Az MTA INTÉZET reláció 1 normálformáju alakját a 6. ábra mutatja:

| M# | NÉV | TELEPHELY | CIM |
|---|---|---|---|
| M1 | MTA SzTAKI | FŐ ÉPÜLET | KENDE U |
| M1 | MTA SzTAKI | SZÁMITÓKÖZPONT | VICTOR H |
| M1 | MTA SzTAKI | VÁR | URI U. |
| M2 | MTA KFKI | FŐ ÉPÜLET | KONKOLY |
| M2 | MTA KFKI | CSILLAGVIZSGÁLÓ | KONKOLY |




6. ábra

A továbbiakban az 1 normálformáju relációt egyszerüen normalizáltnak nevezzük.

**Megjegyzés.** A relációs adatbázis modell csak a normalizált (1NF) relációkat tartalmazhat. Ellenkező esetben felvetődik az összes olyan probléma, ami a hierarchikus adatbázis modellal kapcsolatos, mivel egy normalizálatlan reláció analóg a hierarchikus rekordok file-jával.

Most már pontosabban meg tudjuk határozni a relációs adatmodellt. Felhasználói szempontból nézve ez nem más mint az osztályozott foku ($n$) normalizált relációk (1NF) időben változó halmaza. Az idő-variáns fogalmat azért kell bevezetnünk, mert igy valósitható meg a szám $n$-esek módositása, törlése, beszurása stb. A szám $n$-es valójában egy rekodnak felel meg.

Könnyü belátni, hogy a hagyományos terminológia szerint a reláció megfelel egy (homogén) file-nak, a rekordnak egy szám $n$-es stb. Vagyis a hagyományos logikai adatállományoknál létező fogalmak megfelelői (rekordtipus, rekordelőfordulás, tulajdonság, konkrét érték, azonositó stb.) a relációs adatmodellnél is megtalálhatók. Ezek a megfeleltetések csak közelitések.

Csak néhány példát emelünk ki az eltérések közül:
A file-ok rendszerint szekvenciálisak;
előfordulhat, hogy több rekordnak is ugyanaz az azonositója; szerepelhetnek adatcsoportok; stb.

**Definició.** Az 1 normálformájú *R* reláció 2 normálformáju akkor és csak akkor, ha minden olyan értelmezési tartomány, amely nem elsődleges kulcs vagy annak a része, funkcionálisan függ az elsődleges kulcstól.

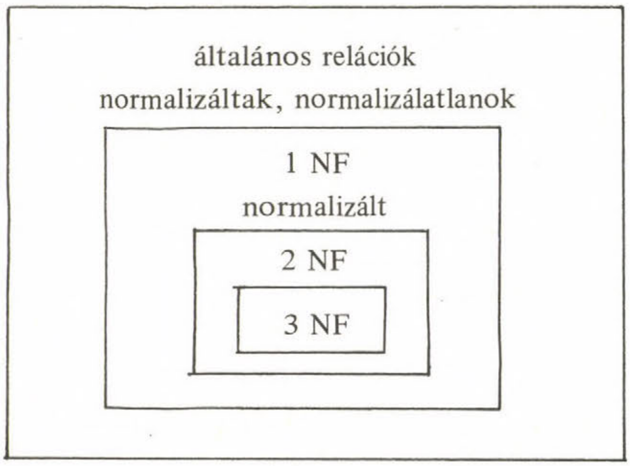
**Megjegyzés.** Könnyü belátni, hogy ha az elsődleges kulcsot egyetlen egy értelmezési tartomány határozza meg, azaz maga a kulcs egyszerü, akkor minden 1 normálformáju reláció 2. normálformáju is. Ez következik abból az egyszerü tényből, hogy ilyenkor a függőség mindig funkcionális teljes függőség is.

Ez a megjegyzés értelmében a DIÁK és TANÁR relációk második normálalakuak. Könnyü belátni, hogy bár a TANÁR-DIÁK reláció kulcsa összetett, a reláció mégis második normálformában van.

**Definició.** Második normálformában lévő *R* reláció harmadik normálformában van, akkor és csak akkor, ha nem tartalmaz tranzitiv függőséget.

5. **A normálformák előnyei**

A Codd által definiált normálformák hierarchiáját a 7. ábra mutatja be.




7. ábra

Az eddig elmondottakat egy példán keresztül illusztráljuk, s közben rámutatunk a normálfotmák jelentőségére, illetve előnyeire.

Egy boltban minden nap zárás után a következő adatokat jegyzik fel (NAPI HELYZET): mennnyi volt az aznapi bevétel (ÖSSZEG), s ezt egy azonositóval (DÁTUM)-al is ellátják. Milyen árukból (ÁRUNÉV) mennyit adatak el (DARAB). Az áru ÁRUKOD-al el van látva, s fel van tüntetve az ára (EGYSÉGÁR). Továbbá regisztrálják azt, hogy történt-e pénz szállitás vagy nem. Amennyiben a napi bevétel (ÖSSZEG) 10.000,-Ft alatt van, akkor nem történik pénzszállitás a bankba, ellenkező esetben azonban igen.

NAPI HELYZET (DÁTUM, ÁRUKOD,

ÁRUNÉV, EGYSÉGÁR, ÖSSZEG, SZÁLLITÁS, DARAB)

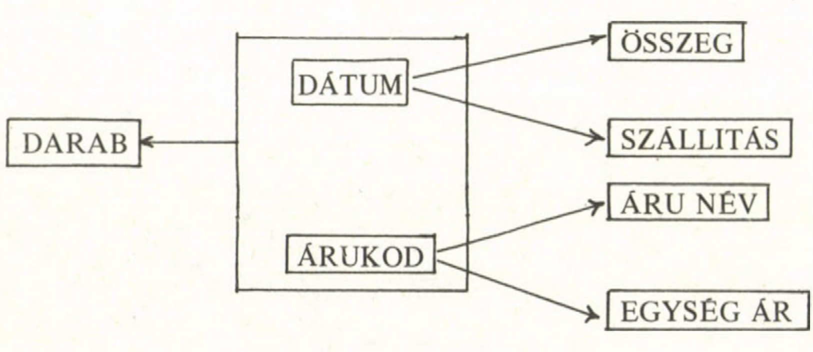
Könnyü belátni, hogy a NAPI HELYZET reláció sorait sem a DÁTUM, sem pedig az ÁRUKOD konkrét értékei nem azonositják egyértelmüen, mivel naponta több árut adhatnak el.

Ebből következik az is, hogy a relációk normalizátlan. (Az ÁRUKOD, ÁRUNÉV, EGYSÉGÁR, DARAB értelmezési tartományokban nem biztos, hogy elemi érték szerepel.) Ha ezekben a nem elemi értékeket tartalmazó értelmezési tartományokban az értékeket egymás alá irjuk és a DÁTUM, ÖSSZEG, SZÁLLITÁS megfelelő értékeit értelemszerüen duplikáljuk., akkor a NAPI HELYZET reláció első normálformájú lesz, amelynek összetett kulcsa lesz (DÁTUM, ÁRUKOD).

| DÁTUM | ÁRUKOD | ÁRUNÉV | EGYSÉGÁR | ÖSSZEG | SZÁLLITÁS | DARAB |
|---|---|---|---|---|---|---|
| 19770915 | A1 | RÁDIÓ | 1000 | 10000 | IGEN | 1 |
| 19770915 | A3 | TELEVIZIÓ | 4000 | 10000 | IGEN | 2 |
| 19770915 | A6 | KERÉKPÁR | 1000 | 10000 | IGEN | 1 |
| 19770916 | A2 | MOSÓGÉP | 3000 | 6000 | NEM | 2 |
| 19770917 | A1 | RÁDIÓ | 1000 | 15000 | IGEN | 3 |
| 19770917 | A4 | MAGNETOFON | 2000 | 15000 | IGEN | 5 |
| 19770917 | A9 | FÉNYKÉPEZŐ | 2000 | 15000 | IGEN | 1 |

8. ábra

A 9. ábrán a NAPI HELYZET reláció funkcionális függőségeit ábrázoljuk.




9. ábra

**Megjegyzés.** Vegyük észre, hogy a funkcionális nem teljes függőség ellenőrzése nem más, mint annak a ténynek a megvizsgálása, hogy a feltételezett kulcs valóban elsődleges kulcs-e.

A NAPI HELYZET reláció nem második normálformáju, mivel bizonyos értelmezési tartományok függnek az összetett kulcs egyes részeitől is.

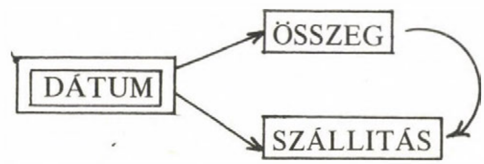
Álatlában nem állunk meg az 1. normálformánál, hanem a relációkat tovább bontjuk, s ezért van, mert az 1. normálalak több "rendellenességet" tartalmaz. Ezeket a kapcsolatokat szüri ki a 2. illetve 3. normálforma. Ezeknek a bevezetése "megtisztitja" az 1. normálformát a három legjelentősebb nehézkeségtől: a törlési-, a hozzáadási- és a felujitási adatfüggőségtől.

Ezekre az adatfüggőségekre még visszatérünk a 2. illetve 3. normálformára bontás után.

Könnyü belátni, hogy a NAPI HELYZET relációt 3 darab 2. normálformáju relációra bonthatjuk, amelyeket a 10. ábra illusztrál.

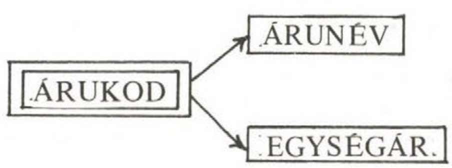
BEVÉTEL (DÁTUM, ÖSSZEG, SZÁLLITÁS)

| DÁTUM | ÖSSZEG | SZÁLLITÁS |
|---|---|---|
| 19770915 | 10000 | IGEN |
| 19770916 | 6000 | NEM |
| 19770917 | 15000 | IGEN |



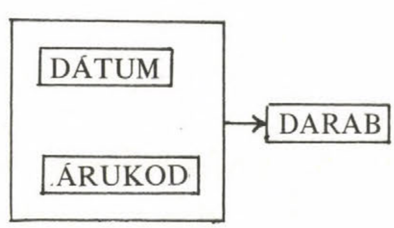

ÁRU (ÁRU-KOD, ÁRU-NÉV, EGYSÉGÁR)

| ÁRU-KOD | ÁRU-NÉV | EGYSÉGÁR |
|---|---|---|
| A1 | RÁDIO | 1000 |
| A2 | MOSOGÉP | 3000 |
| A3 | TELEVIZIO | 4000 |
| A4 | MAGNETOFON | 2000 |
| A6 | KERÉKPÁR | 1000 |
| A9 | FÉNYKÉPEZŐGÉP | 2000 |




MENNYISÉG (DÁTUM, ÁRU-KOD, DARAB)

| DÁTUM | ÁRUKOD | DARAB |
|---|---|---|
| 19770915 | A1 | 1 |
| 19770915 | A3 | 2 |
| 19770915 | A6 | 1 |
| 19770916 | A2 | 2 |
| 19770917 | A1 | 3 |
| 19770917 | A4 | 5 |
| 19770917 | A9 | 1 |




10. ábra

Az ÁRU és MENNYISÉG reláció 3. normál formáju is, de a BEVÉTEL reláció a tranzitiv függőség miatt még nem az. Ha meg akarjuk szüntetni, akkor 2 darab relációra kell bontani (11. ábra), amelyek már 3. normál formában lesznek.

NAPI BEVÉTEL (<u>DÁTUM</u>, ÖSSZEG)

| DÁTUM | ÖSSZEG |
|---|---|
| 19770915 | 10000 |
| 19770916 | 6000 |
| 19770917 | 15000 |

$\boxed{\text{DÁTUM}} \rightarrow \boxed{\text{ÖSSZEG}}$

SZÁLLITÁS (ÖSSZEG, SZÁLLITÁS)

| ÖSSZEG | SZÁLLITÁS |
|---|---|
| 10000 < | NEM |
| 10000 ⩾ | IGEN |

$\boxed{\text{ÖSSZEG}} \rightarrow \boxed{\text{SZÁLLITÁS}}$

11. ábra

Végül megállapithatjuk, hogy a NAPI HELYZET relációt 4 darab 3, normálformáju relációra tudtuk bontani (ÁRU, MENNYISÉG, NAPI BEVÉTEL, SZÁLLITÁS), amelyek tartalmazzák mindazt az információt melyet a NAPI HELYZET reláció tartalmazott.

A relációs adatbázis legfontosabb pontja a normalizálás. Legjobban ezt a tényt az bizonyitja, hogyha megengednénk normalizálatlan relációk használatát, akkor megnyitnánk a kaput az összes olyan probléma előtt, ami a hierarchikus közelitéssel kapcsolatos, mivel a normalizálatlan reláció analóg a hierarchikus rekordok több vagy kevesebb fileval.

Könnyen belátható, hogy a funkcionális függetlenséget nem tartalmazó második normálforma lényeges előnyökkel rendelkezik az első normálformávall szemben.

Nézzük meg ezek közül elsősorban azokat amelyek kapcsolatban vannak az adatfüggőséggel.

A felfrissitési adatfüggőség problémáját az alábbi példa szemlélteti. Tételezzük fel, hogy hónap közben visszamenőleg modositani kell a NAPI HELYZET relációban az ÁRUKOD tartomány konkrét értékeit. Ez a feladat jelentős feldolgozási igényt jelent, mivel a NAPI HELYZET reláció minden egyes sorát el kell olvasni, s meg kell vizsgálni, hogy az ÁRUKOD aktuális értéket kell-e módositani. Nyilvánvaló, hogy ez a megoldás lényegesen több munkát és időt igényel, mint a második normál formában levő ÁRU relációban történő változtatások átvezetése.

Vegyük észre, azt, hogy ha a modositandó tulajdonság idegen kulcs, akkor az adatkapcsolatok helyessége is veszélyben van, mivel ez az érték az adatbázis más relációjában is szerepel.

Tehát megállapithatjuk, hogy a második normálformában lévő relációkban minden egyes módositás lényegesen könnyebben és gyorsabban hajtható végre, mint első normál formában.

A törlési adatfüggőséggel kapcsolatos problémákat azonnal láthatjuk, hogyha a NAPI HELYZET relációból az 19770916A2 kulcsu sort. Ez ugyanis azt jelenti, hogy a mosógépre vonatkozó alapvető információinkat (ára, kódja) is elveszitettük, pedig a mosógépre vonatkozó adatokat még tárolunk. Ezzel szemben az ugyanilyen azonositóju sor törlése a MENNYISÉG relációból nem jelenti a mosógépre vonatkozó alapvető információk elvesztését, mivel az az ÁRU relációban került tárolásra.

Megállapitjuk tehát, hogy a második normálforma használatánál – szemben az első normálformával – törlés esetén nem vész el lényeges információ, amely az első normálforma esetén esetleg elveszett volna. A hozzáadási adatfüggőség a szemléltetésére vegyük azt a példát, hogy a NAPI HELYZET relációhoz hozzá kell adni egy új információt a lemezjátszóról (ÁRU-KOD: A8, EGYSÉG: 2000) amit az árusitók már tudnak mivel árulják a lemezjátszót, de a NAPI HELYZET relációba nem tudjuk beirni, mert még senki nem vett egyetlenegy lemezjátszót sem, s igy nincs elsődleges kulcsunk erre a szám-7-es-re.

A második normálformában levő ÁRU relációban minden további nélkül beilleszthető a lemezjátszóval kapcsolatos új információink.

Megállapithatjuk, hogy a második normálforma estén a relációk konzisztensen bővithetők anélkül, hogy az adatbázisba hamis adatokat vinnénk be.

Vegyük észre, hogy a két utóbbi adatfüggőséggel kapcsolatos problémáink szorosan összefüggnek a relációink terjedelmével.

A második normálforma a felsorolt adatfüggőségi előnyein túl számos más előnnyel is rendelkezik, ezek közül pl. jelentős az, hogy egyszeresen szerepelnek azok az adatok, amelyek az első normálforma esetén feleslegesen ismétlődnek (ÁRUNÉV, EGYSÉGÁR). Mivel a második normálforma bevezetésével megszünt az ilyen eredetü redundancia, ezért a relációink együttes mérete csökkent (általában!), az eredeti relációval szemben. Igy kisebb a veszély az inkonzisztens adatok tekintetében.

Könnyü belátni, hogy a harmadik normálforma előnyei a második normálformával szemben hasonlóak az előbb elmondotakhoz. Ez azzal a ténnyel magyarázható, hogy a harmadik normálforma megoldja a második normálformában levő tranzitiv függőségek eliminilását is.

## I r o d a l o m

[1] D.L. Childs: Description of a Set-Theoretic Data Structure. Proc. FJCC, Vol. 33, Part 1, December 1968. (557-564).

[2] E.F. Codd: A relational model of date for large shared data banks, CACM 13 (1970) 377-387.

[3] E.F. Codd: Normalized data base structure: A brief tutorial, Proc. 1971. ACM-SIGFIDET Workshop on Data Description, Access and Control.

[4] E.F. Codd: Further normalization of the data base relational model, Courant Computer Science Symposia 6 "Data Base System", (Prentice Hall, Englewood Cliffs, N.J., 1971) 33-64.

[5] C.J. Date: An introduction to database systems, Addison-Wesley, Reading, Mass. 1975.

[6] Halassy Béla: Adatbázisok számitógépes kezelésének alapvatő kérdései. SZÁMOK 1978.

[7] Gyürki József: Modern számitógépes adatbázis kezelő software rendszerek és nyelvek. BME Jegyzet 1976.

## S u m m a r y

The relational data base model

János Demetrovics

In the presert paper the author gives a survey on relational data base structures which is mainly base on Codd's approach.

## Р Е З Ю М Е

Релационная модель баз данных

Янош Деметрович

В настоящей работе лается некоторый обзор баз данных о нормалформах по работам E.F. Codd.

MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények** 20/1978.

# EGY SZÉLES KÖRBEN ALKALMAZHATÓ PROGRAMOPTIMALIZÁLÁSI MÓDSZER

Ruda Mihály

## 1. Bevezetés

A számológépgyártásban és felhasználásban sokféle optimalizálási törekvésnek lehetünk tanui. Egy adott hardware-konfiguráció hatékonyságát a gép software-je is nagymértékben befolyásolja, a software fejlesztését pedig a gép felhasználási köre határozza meg. A software-fejlesztés – pl. a magasszintű nyelvek implementásása, a fejlett operációs rendszerek, célnyelvek, felhasználói programrendszerek, programkönyvtárak – célja általában a gépfelhasználás hatásfokának javitása (az emberi munkaerő és a hardware eszközök minél jobb kihasználása).

Vannak azonban olyan tényezők is amelyek a feladatok gépigényének (idő, tároló) minimalizálását követelik. Ez a kérdés különösen élesen vetődik fel akkor, amikor a rendelkezésre álló gépkapacitás határát érintő feladatokat kell megoldani.

Ilyen célt szolgálnak például a forditóprogramok optimalizálási törvényei ( [7], [10]), vagy a multiprogramozással kapcsolatban az optimális kiszolgálási stratégiák kidolgozása ([1], [3]). Az utóbbi kérdéskörön belül lényeges szerepet játszanak a statitisztikai és valószinűségszámitási meggondolások ([2]).

Ezeken az optimalizálási törekvéseken túl, más módon is javitható egy számológép működésének hatásfoka. Ide soroljuk azt a feladatot is, amely egy állandó jelleggel használt program (programrendszer) "általános megfogalmazása" és az esetenként futó speciális feladat között lévő lényeges különbség felszámolását jelenti. Részletesebben a következőkről van szó:

Azok a programok amelyek egy általános feladatkör megoldására készülnek (pl. lineáris egyenletrendszerek megoldása), általában paraméterek megadásával vezérelhetők a kitűzött feladat elvégzésére. Ezek a programok esetenként a futás közben sokszor ismétlődő eljárásokon belül is ujra és ujra megvizsgálják a vezérlő paramétereket, illetve ujra és ujra ismétlődő számitásokat végeznek velük.

A probléma onnan ered, hogy a programok forditási és futtatási fázisa – a szokásos compiler-eknél – élesen szétválik. Ezért olyan adatok, amelyek konkrét esetekben állandó értékek, változóként szerepelnek, sőt néha a program bonyolult eljárásokkal feleslegesen hozza létre őket ujra és ujra. Ugyanigy egyes magasszintű nyelveken (pl. FORTRAN, COBOL) nem valósitható meg a (valóban) dinamikus memóriakihasználás ( az ALGOL lehetővé teszi változó méretű tömbök alkalmazását). Egyszerü programoknál a felhasználó (a programozó) kézi beavatkozással – pl. FORTRAN programban egy DIMENSION kártya cseréjével – is módosithatja a programot, a soronlévő feladat igényeinek megfelelően. Egy bonyolult programrendszernél azonban semmiképpen sem lehetséges egy futásonként ismétlődő, kézi beavatkozással történő programváltoztatás – még a legegyszerűbb formában sem. Ez méginkább igy van, ha maga a módositás is bonyolult.

A helyes megoldás egy olyan operációs rendszerváltozat kialakitása, amely az általános használhatóság mellett lehetővé teszi azt, hogy a programok forditása egyben egy "átdolgozás" is legyen, sőt azt is, hogy futás közben módosithassuk a már leforditott programot is (ugyanugy mint a gépikód programoknál). Átdolgozás alatt itt azt értjük, hogy a rendszer, a felhasználó által adott paraméterek függvényében, futásonként más és más módon értelmezi a programot.

Az oprációs rendszerek néhány ujszerű feladaton túl (mint pl. az előbb már emlitett multiprogramozás futásütemezése) elsősorban a hagyományos "kézi" gépkezelési feladatokat (forditás inditás, programbetöltés, tárolók nyitása, zárása, szalagok vezérlése, stb.) látnak el. Felhasználói munkák kezelése más módon és más tipusu területeken is történhet mint a hagyományos "kézi" operátori munka. A következőkben egy ilyen lehetőséggel foglalkozunk. Egy olyan eljárás alkalmazását mutatjuk be, amely a szokásos operációs rendszerek segitségével is lehetővé teszi a programok automatikus átdolgozását (esetenkénti ujraértelmezését) – pl. optimalizálási céllal.

## 2. Az alkalmazott módszer leirása

A módszer, amelynek alkalmazására a 3. pontban példákat is adunk, a következő: Az optimalizáláni kivánt programot úgy készitjük el, hogy az egyes futásoknál az aktuális paramétereket egy előkészitő program illeszti a futó programba. A feladat tehát két program irása: az első az előkészitő program, amely a futó programot az aktuális paramétereknek megfelelően összeállitja; a második programnak (a tulajdonképpeni feladatot ellátó programnak) csak az állandó részeit készitjük el előre, a változó részeket az előkésztitő program illeszti be. A futó program igy mindig egy a pillanatnyi helyzetnek megfelelő optimális változatban működhet.

A módszer hatékonyságát a következő szakaszban bemutatott példákkal illusztráljuk. Ezekben a példákban bemutatott eljárásokat a szerző sikeresen alkalmazta egy Honeywell Bull 66/60-as gépen készülő nagyméretű információs rendszer megvalósitásánál. Az egyes optimalizálási eljárások programozásában GÁL ANNA, RATKÓ ISTVÁN és VASS RÓZSA vett részt. A rendszer egy vázlatos leirása [13]-ban található. A módszerrel kapcsolatos első eredmények az MTA SzTAKI-ban működő CDC 3300-as gépen születtek 1975 tavaszán (ld. [4], [5], [6]).

Az itt bemutatott módszerrel kapcsolatban két dolgot fontos tisztázni: 1. az előkészitő programok előállitása nem túl bonyolult-e, 2. a program (rendszer) felhasználóját nem terheli-e feleslegesen az uj technikai alkalmazása.

Egy közvetve adódó de igen jelentős előnyre kell még felhivni a figyelmet. Az esetenként ujra és ujra összeállitott program mindig egy-egy speciális feladatot old meg, tehát az általános célhoz viszonyitva egyszerű és könnyen áttekinthető lesz a program. Ez pl. a hibakeresést (elsősorban a programirás fázisában, de a felhasználásnál is) rendkivüli módon megkönnyiti.

Most bemutatunk néhány példát. Hangsúlyozni kell, hogy a vizsgált módszert olyan esetekben célszerű alkalmazni, amikor egy sokszor ismételt futtatásra szánt nagy hely- és időigényű programot készitünk. Csak a futásiidő csökkentését illusztráló példákat közlünk. A tárolóigény csökkentésénel lehetőségét például azzal biztosithatjuk, hogy lehetővé tesszük az adattömbök méretének futásonkénti (vagy futás közbeni) változtatását (úgy mint pl. az ALGOL-ban). Természetesen egy előkészitő (programszerkesztő) program alkalmazásával az adatterületek nagyságának minimalizálásán túl még más lehetőségek is vannak a tárolóigény csökkentésére, például az éppen futó programváltozatban felesleges utasitások elhagyásával.

## 3. Példák

1. Az itt következő példával csak röviden foglalkozunk, a probléma részletesebb leirása a [8], [11] és [12] dolgozatokban található meg. A feladat a következő: Előre nem ismert, bonyolult logikai kifejezések kiértékelését kivánjuk megoldani. A feladatot még az is nehezíti, hogy a kifejezésekben szereplő változók értékét csak igen nagyméretű értéktáblázatokkal tudjuk megadni.

Változó formájú és tartalmú logikai kifejezések programba való épitése úgy is megoldható, hogy a felhasználó – mondjuk FORTRAN nyelven (FORTRAN program esetén) – leirja a feltételt, mindig az éppen szükséges formájában, és egy szerkesztő eljárással a megfelelő helyre illeszti. Ez tuladonképpen már az általunk javasolt módszer egy változata. Sokkal bonyolultabb a probléma akkor, ha a változók igen sokféle értéket vehetnek fel, esetleg olyan értékek szerepelnek amelyek más eljárások eredményeként adódnak. Ilyenkor bonyolult logikai kifejezések és terjedelmes értéktáblázat rendszerek leirását és vizsgálatát kell megoldani. Ebben az esetben az előkészitő program fő célja az aktuális logikai kifejezések lehető leggazdaságosabban kiértékelhető változatának összeállitása.

2. Ebben a példában azt az általános elvet valósitjuk meg, mely szerint, ha egy függvényt egy ismételt eljárás folyamán sokszor kell kiszámitani, akkor az ismételt számitások helyett egy értéktáblázatot készitünk, és ezután ebből másoljuk ki a szükséges függvényértékeket. (Tulajdonképpen ez az elv lett alkalmazva az 1. példa esetében is – ld. pl. [12]).

A vizsgált feladat a következő: Fix rekordok kijelölt adataiból készitünk gyakoriságtáblázatot (pl. egy populáció kor és nem szerinti megoszlását határozzuk meg). Ha általános formában kivánjuk megoldani a feladatot – akárhány dimenziós táblázatot készithetünk a rekord tetszőleges adatiból – akkor egy bonyolult, többszörös indexezésekkel müködő programot kell irni. A táblázat értékeinek gyüjtése, vagyis az aktuális táblaindex beállitása pl. a következőképpen irható le:

```
      IND = L(S(M))- KA(M) + 1
(1)   DO 1 I = 1,N
    1 IND=IND + T(I) * (L(S(I)) - KA(I)),
```

ahol M a táblázat változóinak száma,

N = M – 1,
L a rekodelemeket tartalmazó vektor,
S a táblázat változóinak (a rekordon belüli) sorszámait tartalmazó vektor,
KA a táblaadatokra adott alsó korlátok vektora,
T(I) az I-edik szintű résztáblák terjedelme,
azaz T(M) = 1, T(M – 1) = KF(M) – KA(M) + 1, stb.,
KF a táblaadatok felső korlátainak vektora,
IND a számitandó index.

Ezt a programrészt a következő FORTRAN nyelvű változatokkal is megoldhatjuk, ha már ismerjük az elvégzendő feladatot, amely mondjuk egy háromdimenziós táblázat készitése:

(2) IND = T1 * L1 + T2 * L2 + L3 + D,

ahol az előző jelölések szerint

T1 = T(1),
T2 = T(2),
L1 = L(S(1)),
L2 = L(S(2)),
L3 = L(S(3)) és
D = 1 – KA(3) – T(2) * KA(2) – T(1) * KA(1).

Ha a futtatás előtt ismerjük a feladat paramétereit, akkor T1, T2 és D kiszámitható, az L1, L2, L3 változónév pedig hozzárendelhető a rekord kivánt elemeihez. Ezeket a feladatokat (a számitásokat és a hozzárendeléseket) az előkészitő program végzi el.

A (2)-höz hasonlóan használhatjuk az

(3) IND = T1(L1) + T2(L2) + T3(L3) . . .

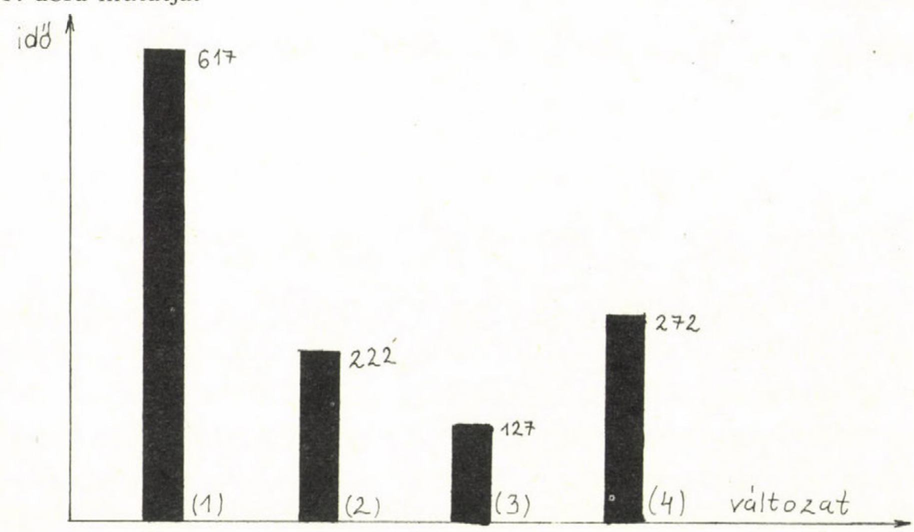
utasitást, ha az L1, L2, L3, . . . adatok lehetséges értékeinek megfelelő indexnövekményeket – azaz a T(I) * (L(S(I)) – KA(I)), (I = 1, . . . ,N) és az L(S(M)) – KA(M) + 1 értékeket – előre elhelyezzük a T1, T2, T3, . . . táblázatokban.

Ha az adatok (rekordelemek) között negativ értékek is előfordulnak, vagy ha a KA(I) alsó korlátoknál kisebb értékeknek nem kivánunk feleslegesen helyet fenntartani, akkor a következő megoldást használhatjuk:

J1 = L1 – KA1
J2 = L2 – KA2
J3 = L3 – KA3
IND = T1(J1) + T2(J2) + T3(J3) . . . ,

ahol KA1 = KA(1), stb. (Meg kell jegyezni, hogy az első három utasitást EOUIVALENCE utasitások segitségével kiküszöbölhetjük.)

Jól látható, hogy a (2) – (4) programrészlet – főleg a (3) – lényegesen gyorsabb mint az (1) változat. A tényleges sebességkülönbségek azonban függnek a felhasznált számológéptől is. A négyféle programvariáció futásidő arányát, egy Honeywell Bull 66/60-as gépen történt kisérlet alapján az 1. ábra mutatja.




1. ábra

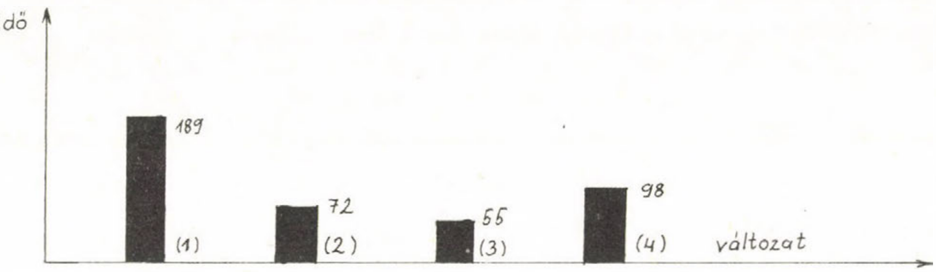
Az ábrán az időértékek tizezred-órában adottak. A kisérleti programban M = 10 volt, és az indexszámitási eljárást egymilliószor ismételtük. Az eredeti (1) változat és a leggyorsabb (3) változat időaránya

$$617/127 \sim 4.9.$$

Mivel a közölt értékek kisérleti adatok, ezért (a pillanatnyi futási környezet függvényében) ingadozhatnak, de ezek az ingadozások nagyságban messze elmaradnak a fenti, közel ötszörös sebességnövekedéshez viszonyitva.

Az időarány függhet az M értékétől is. M = 3-ra (ugyancsak egymillió ismétléssel) a következő időadatokat kaptuk (ld. 2. ábra). Itt a két szélső időérték közti arány:

$$189/55 \sim 3.4.$$




2. ábra

3. A most soronkövetkező példa egy még nagyobb mértékű futásidő csökkenést mutat be.

A legtöbb esetben a feldolgozandó adatok karakteres formában kerülnek a felhasználó kezébe, de a szükséges eljárások lényegesen gyorsabban hajthatók végre bináris adatokkal. Iyen esetekben konverziót alkalmazunk. A konverzió általában időigényes eljárás. Végrehajtási ideje nagymértékben függhet attól, hogy a konverzió egy külső, tárólóból való olvasással párosul-e, vagy csak a belső tárolóban történik. Mindkét esetben figyelembe kell venni a felhasználatlan karakterek szerepét is. A külső tárolón, vagy a memóriában hézagosan elhelyezett karaktermezők konverziója általában több időt igényel mint a tömören elhelyezetteké. Néhány példa egy Honeywell Bull 66/60-as gépen mágnesszalagról FORTRAN programmal beolvasott 150 ezer rekord esetén (itt és az előzőkben is, az időértékek a központi egység által felhasznált időt mutatják):

| FORTRAN formátum | idő(órában) |
|---|---|
| 77X,I2,I2 | 0.1508 |
| A80 | 0.1549 |
| 4X,I2,17X,I2 | 0.0951 |
| 22X,I1,84X,I3,31X,I2 | 0.2421 |

1. táblázat

A konverzió sebességének növelésére alkalmazott módszer az itt bemutatott formájában csak nem negativ egész értékek konvertálására használható (igaz, hogy adatfeldolgozási feladatoknál ez a leggyakoribb eset). A módszert kétjegyű szám esetére mutatjuk be. A konverziót most nem kiséri beolvasási művelet.

A konvertálandó két karakter az N1, N2, . . . szavakban foglal helyet, mondjuk az

N5 szó 6-11 és 12-17 bitjein (egy karakter 6 bit). Ekkor az M szóba irányuló konverzió a következőképpen oldható meg:

I1 = FLD(6,6,N5)
I2 = FLD(12,6,N5)
M = M1(I1) + I2

ahol M1(I) = I * 10, és az FLD függvény (Honeywell FORTRAN könyvtári eljárás) értéke a következő: FLD(a,b,c) a c szó b bitje a-adik bittől kezdve. Ez a b drb. bit kerül a baloldalon álló változó alsó helyiértékeire. Az utasitásokat itt is egy előkészitő program állitja össze.

Ezt az eljárást a hagyományos konverzióval összehasonlitva, 300 ezer ismétlés esetén a következő adatokat kaptuk (2. táblázat):

| konverziós forma | idő |
|---|---|
| 30X,I2 formátum | 0.1169 |
| a gyorsitott eljárás | 0.0014 |

2. táblázat

Az arányokat (majdnem százszoros különbség) az előzőkhöz hasonló diagramon nem is tudjuk bemutatni. A sebességnövekedés egyik nyilvánvaló oka az, hogy a bemutatott eljárás független a felhasználatlan karakterek számától.

**4. Összefoglalás**

A 2. pontban javasolt általános módszer hasznossága a bemutatott példák alapján nyilvánvaló. A figyelmet arra kell még felhivni, hogy az előzőkben példákon keresztül bemutatott eljárások (az előkészitő program alkalmazása) nagyméretű feladatoknál nem csak hasznosak, de sokszor nélkülözhetetlenek is. A módszer gyakorlati alkalmazhatóságát a 2. pontban emlitett adatfeldolgozó rendszerben történt sikeres felhasználása bizonyitja.

# Irodalom


[1] Aho A.V. – Denning P.J. – Ullmann J.D.: Principles of optimal page replacement, J. Assoc. Comput. Mach., 18. (1971).

[2] Benczúr A. – Krámli A. – Pergel J.: On the Bayesian approach to optimal performance of page storage hierarchies, Acta Cybernetika, 3., (1977).

[3] Coffman E.G. – Kleinrock L.: Computer Scheduling Methods and their Countermeasures, Proceedings, AFIPS (1968), SJCC.

[4] Csukás A-né, – Greff L. – Krámli A. – Ruda M.: A kórházi morbiditási vizsgálat számitógépes feldolgozásának tapasztalatai és továbbfejlesztése, Számitástechnikai és kibernetikai módszerek alkalmazása az orvostudományban és a biológiában, 5. Kollokvium, Szeged, (1974).

[5] An approach to the hospital morbidity data system development in Hungary, Symposium on Medical Data Processing, Toulouse, (1975).

[6] Lekérdező rendszer kórházi morbiditási vizsgálat anyagára, Számitátechnikai és kibernetikai módszerek alkalmazása az orvostudományban és a biológiában, 6. Kollokvium, Szeged, (1975).

[7] Finkelstein M.: A Compiler Optimization Technique, Computer Journal, 2., (1968).

[8] Krámli A.,– Ratkó I. – Ruda M. – Soltész J.: A statisztikai adatfeldolgozás matematikai és számitástechnikai problémái, MTA SzTAKI Tanulmányok, 70/1977.

[9] Knuth J.D.E.: The art of computer programing, Sorting and Searching (3. kötet), Addison-Wesly, London, California, (1973).

[10] Nievergelt J.: On the Automatic Simplification of Computer Programs, CACM, 8., (1965).

[11] Ratkó I.: Egy számitástechnikai eszköz bonyolult logikai kifejezések leirása orvosstatisztikai alkalmazásban, Számitástechnikai és kibernetikai módszerek alkalmazása az orvostudományban és a biólogiában, 8. Kollokvium, Szeged, (1977).

[12] Ratkó I.: Bonyolult logikai kifejezések kiértékelésének számitástechnikai és optimalizálási problémái, MTA SzTAKI Közlemények, 20/1978.

[13] Ruda M.: Egy általános információs rendszer kórházi morbiditási adatok feldolgozására, Számitástechnikai és kibernetikai módszerek alkalmazása az orvostudományban és a biológiában, 8. Kollokvium, Szeged, (1977).

## Summary

### A generally applicable program optimizing method

Mihály Ruda

There are various endeavours for effective use of computers, for example automatic program optimization at compile time, computer scheduling methods, etc. This paper examines a computer program optimization method applicable in operating systems too. Examples (for the decrease of compute time) show the effect of the method (cf. fig. 1., 2. and table 2.). The author applied this method successfully in a medical information system. The method works by a preparatory (editor) program which recomposes the source program before each run.



## РЕЗЮМЕ

### ОБ ОДНОМ МЕТОДЕ ОПТИМИЗАЦИИ ПРОГРАММ, ИМЕЮЩЕМ ШИРОКОЕ ПРИМЕНЕНИЕ

Михай Руда

Известны различные способы повышения эффективности ЭВМ, например, автоматическая оптимизация трансляторов, использование различных стратегий при выборе режима работы операционной системы и др.

В данной работе рассматривается метод оптимизации программ, который может быть использован и в операционных системах. Эффективность метода /уменьшение машинного времени/ иллюстрируется на конкретных примерах /см. рис. 1 и 2, а также табл. 2/. Описываемый метод автор с успехом использовал при разработке одной информационной системы. Суть метода: оптимизируемую программу составляет специальная подготовляющая программа /редактор/.

MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények** 20/1978.

# BONYOLULT LOGIKAI KIFEJEZÉSEK KIÉRTÉKELÉSÉNEK SZÁMITÁSTECHNIKAI ÉS OPTIMALIZÁLÁSI PROBLÉMÁI

Ratkó István

Az ÁSzSz HwB gépén történő kórházi morbiditási adatfeldolgozás (ld. [1]) vetette fel a következő feladatot.

Adott egy fix hosszúságú rekordokból álló adatfile. A szakembereket igen gyakran csak speciális feltételeknek elegettevő adatok érdeklik. Ezek a feltételek logikai kifejezések formájában irhatók fel. A feltételekkel kapcsolatban a következőket tudjuk: a) sok van belőlük, b) nem rögzithető előre minden lehetséges feltétel, amire szükség lesz. A programozási nyelv szokásos eszközei nem biztositanak olyan lehetőséget, amellyel ez a két követelmény kielégithető. Cikkünkben mutatunk egy megoldási módszert, arra a nem lényegtelen szempontra is ügyelve, hogy a feltételvizsgálat ideje minél kisebb legyen.

## 1. A modell

A rekord álljon $N$ adatelemből, az $i$-edik adatelem értékét jelölje $\tau_i$, $\tau_i$ lehetséges értékeinek halmazát pedig $H_i$. ($H_i$ lehet pl. egy intervallum, de lehet bonyolultabb halmaz is). Valamely konkrét adatfeldolgozásnál az adatfile bizonyos feltételeknek elegettevő rekordjaira van csak szükség. Hogyan irjunk fel egy az ezen rekordokat kiválasztó logikai kifejezést? A kifejezés akkor és csak akkor legyen igaz, ha a rekordra szükségünk lesz. Tegyük fel, hogy a logikai kifejezésben az $i_1, i_2, \ldots, i_M$ sorszámú adatelemekre vonatkozó feltételek szerepelnek. A $k$-adik adatelemmel kapcsolatos feltétek igy néznek ki:

$$(\tau_k \in A) \quad \text{ahol} \quad A \subset H_k .$$

A $(\tau_k \in A)$ itélet tulajdonképpen $|A|$ számú diszjunkcióból áll; itt $|A|$ az $A$ halmaz elemszáma.

Jelölje $A_{i_k, j}$ $H_{i_k}$ azon részhalmazait, amelyek $(\tau_{i_k} \in A_{i_{k,j}})$ formában szerepelnek a rekordokat kiválasztó logikai kifejezésekben ($j = 1,2, \ldots, r_k$, $k = 1,2, \ldots, M$). Nyilvánvaló, hogy logikai kifejezésünknek legalább

$$\sum_{k=1}^{M} \sum_{j=1}^{r_k} |A_{i_{k,j}}| \quad \text{tagja (konjunkció, diszjunkció vegyesen) van.}$$

Ezt hagyományos módon beépiteni a programba reménytelen, sőt legtöbbször lehetetlen. A fenti tagszám ugyanis többezeres nagyságrendü is lehet a bevezetőben emlitett feladatnál.

## 2. A módszer

Mit tehetünk? Redukáljuk a kifejezésben szereplő változók számát. Ezt a következő egyszerü ötlettel érhetjük el.

Definiáljuk a $Z_{i_{k,j}}(s)$ függvényt a következőképpen:

$$Z_{i_{k,j}}(s) = \begin{cases} 0, \text{ ha } s \epsilon A_{i_{k,j}} \\ 1, \text{ ha } s \notin A_{i_{k,j}} \end{cases} \qquad (s \epsilon H_k)$$

Ezzel elértük, hogy a $(\tau_k \epsilon A_{i_{k,j}})$ $|A_{i_{k,j}}|$ számú diszjunkció helyett az egyetlen $Z_{i_{k,j}}(\tau_k) = 0$ tag áll, ugyanis nyilvánvaló, hogy $(\tau_k \epsilon A_{i_{k,j}})$ akkor és csak akkor igaz, ha $Z_{i_{k,j}}(\tau_k) = 0$ igaz. Megjegyezzük, hogy kifejezésünk most legalább $\sum_{k=1}^{M} r_k$ tagból áll.

Bizonyos mértékü egyszerüsitést már elértünk, ha azonban a logikai kifejezés kiértékelési idejét, s igy a futási időt csökkenti akarjuk, további meggondolásra van szükségünk.

A rekordokat kiválasztó $L$ logikai kifejezésünket hozzuk diszjunktiv normálalakra:

$L = L_1 V L_2 V L_3 V \ldots V L_{d-1} V L_d$, ahol tehát $L_i$ konjunkciókból áll.

Ez mindig megtehető (ld. [2]).

Bár a következőkben a mondottakat FORTRAN nyelven szemléltetjük, hasonlót állithatunk COBOL nyel alkalmazása esetén.

Nyilvánvaló, hogy az

IF(L.EQ.FALSE) GO TO 2 (a rekord kihagyandó)

utasitás végrehajtása több időt vesz igénybe, mint az

IF($L_1$.EQ.TRUE) GO TO 1
IF($L_2$.EQ.TRUE) GO TO 1
.
.
.
IF($L_d$.EO.TRUE) GO TO 1
GO TO 2
1 szükség van a rekordra
utasitáscsoporté.

Mig ugyanis az első esetben $L$ értékének eldöntéséhez az összes konjunkciót ki kell értékelni, a második esetben a kiértékelés hamarabb befejeződik.

Az IF($L_i$.EQ.TRUE) GO TO 1 tovább bontható.

Legyen evégből: $L_i = L_{i1} \wedge L_{i2} \wedge L_{i3} \ldots \wedge L_{iq_i}$

Az IF($L_{i1}$.EQ.FALSE) GO TO $C_{i+1}$
IF($L_{i2}$.EQ.FALSE) GO TO $C_{i+1}$

(1) IF($L_{i_{qi}}$.EQ.FALSE) GO TO $C_{i+1}$
GO TO 1

$\left.C_{i+1}\right\}$ Mint fent, csak az $L_{i+1}$-et bontva

utasitáscsoport végrehajtása hamarabb fejeződik be, mint az IF($L_i$.EQ.TRUE) GO TO 1 utasitásé.

A $T$ futási idő ujabb csökkentését érhetjük el a $Z_{i_{k,j}}(\tau_k)$.EQ.0 események relativ gyakoriságának ismeretében.

Ehhez a következő minimum feladatot kell megoldani. Az $1,2,3, \ldots, d$ számok egy permutációja legyen $m_1, m_2, \ldots, m_d$, az $1,2, \ldots, q_i$ számoké pedig $n_1, n_2, \ldots, n_{q_i}$ $(i = 1,2, \ldots, d)$. Ha (1)-ben az $L_i$ diszjunkciók sorrendje $L_{m_1}, L_{m_2}, \ldots, L_{m_d}$, továbbá az $L_i$ szétbontásában $L_{in_1}, L_{in_2}, \ldots, L_{in_q}$ a kiértékelések várható száma meghatározható. A kérdés: milyen sorrend esetén lesz ez a várható érték minimális?

### 3. A kiértékelési szám minimalizálása

Tegyük fel, hogy az $L = L_1 V L_2 V \ldots V L_d$ felirásnál $L_k$-ban $(k = 1,2, \ldots, d)$ az $l_k$-adik konjunkció hamis, az előtte lévő konjunkciók igazak, továbbá, hogy $L_{k+1}$ igaz. Ekkor a $\rho(L) = l_1 + l_2 + \ldots + l_k + a_{k+1}$ számot $L$ kiértékelési számának nevezzük, ahol $a_{k+1}$ $L_{k+1}$ konjunkcióinak száma. Ha $L_1$ igaz, akkor $\rho(L) = a_1$

$\rho(L)$ minimalizálására törekszünk.

Először vizsgáljuk azt a kérdést, hogy az

$$L = \ldots V(\ldots L_1 \wedge L_2 \wedge \ldots) \ldots \quad \text{és} \quad \overline{L} = \ldots V(\ldots \wedge L_2 \wedge L_1) \ldots) V \ldots$$

kifejezések kiértékelési számairól mit tudunk mondani.
Vezessük be a következő jelöléseket:

$$L' = \ldots \wedge L_1 \wedge L_2 \wedge \ldots \qquad L'' = \ldots \wedge L_2 \wedge L_1 \wedge \ldots$$

$p = P(L_1 = i)$ (azaz hogy $L_1$ igaz) $q = P(L_2 = i)$

$A' = \{$ $L'$-ben minden $L_1$ előtti konjukció igaz és minden $L'$ előtti diszjunkció hamis , továbbá $L_1 = i$, $L_2 = h$ $\}$

$A'' = \{$ $L''$-ben minden $L_2$ előtti konjukció igaz és minden $L''$ előtti diszjunkció hamis , továbbá $L_2 = i$, $L_1 = h$ $\}$

$$p_1 = P(A'),\ p_2 = P(A'')$$

1. Tétel.

*$\rho(L) \leqslant \rho(\overline{L})$ akkor és csak akkor, ha $p(1-q)p_1 \leqslant (1-p)qp_2$.*

Bizonyitás.

Legyen $\xi_1$ és $\eta_1$($\xi_2$ és $\eta_2$) $L(\overline{L})$ kiértékelési száma $L'$ előtt és után ($L''$ előtt és után) azon feltevés mellett, hogy $L_1 = i$, $L_2 = h$ és $L'$-ben minden $L_1$ előtti konjunkció igaz ($L_1 = h$, $L_2 = i$ és $L''$-ben minden $L_2$ előtti konjunkció igaz ). Jelölje $k$ az $L_1$ előtti konjunkciók számát ($L'$-ben, $L''$-ben), tovább $p'$ annak valószinüségét, hogy $L'$-ben minden $L_1$ előtti konjunkció igaz. Nyilván $p'$ annak valószinüségét is megadja, hogy $L''$-ben minden $L_2$ előtti konjunkció igaz.

Elemi meggondolásokból következik, hogy $\rho(L) \leqslant \rho(\overline{L})$ akkor és csak akkor, ha

$$\begin{aligned} &E(\xi_1 + (k+2)\chi_A + \eta_1\chi_A)p'p(1-q) + \\ &\quad + E(\xi_2 + (k+1)\chi_{A''} + \eta_2\chi_{A''})p'(1-p)q \leqslant \\ &\quad \leqslant E(\xi_2 + (k+2)\chi_{A''} + \eta_2\chi_{A''})p'(1-p)q. \end{aligned}$$

Ez pedig ekvivalens (1)-gyel, ami állitásunkat bizonyitja. Megjegyezzük, hogy ha sem $L_1$, sem $L_2$ nem szerepel az $L'$ előtti diszjunkciókban, akkor (1) a $p \leqslant q$ egyenlőtlenséggel ekvivalens.

Azt vizsgáljuk ezek után, hogy az $L = \ldots VL'VL''V \ldots$ és az $\overline{L} = \ldots VL''VL'V \ldots$ kifejezésekre mit mondhatunk $\rho(L)$-ről és $\rho(\overline{L})$-ról.

Jelölje a illetve $b$ $L'$-ben illetve $L''$-ben a konjunkciók számát. A következő jelöléseket használjuk:

$p_1$: annak valószinüsége, hogy $L'$ minden tagja igaz, $L''$ és az $L'$ előtti tagok mindegyike hamis

$\eta$: $L''$-ben az első hamis tag sorszáma, feltéve, hogy $L' = i$

$p_2$: annak valószinűsége, hogy $L', L''$ minden tagja igaz, $L'$ előtti tagok mindegyike hamis

$p_3$: annak valószinűsége, hogy $L''$ minden tagja igaz, $L'$ és az $L''$ előtti tagok mindegyike hamis

$\xi$: $L'$-ben az első hamis tag sorszáma, feltéve, hogy $L'' = i$

2.Tétel.

*$\rho(L) \leqslant \rho(\overline{L})$ akkor és csak akkor, ha*

(2) $$ap_2 + p_3 E\xi \leqslant p_1 E\eta + bp_2$$

Bizonyitás.

Legyen

$\xi_1$: $L$ esetén $L'$-ig ($\overline{L}$ esetén $L''$-ig) a kiértékelt tagok száma azon feltétel mellett, hogy $L' = i$

$\xi_2$: $L(\overline{L})$ esetén $L'$-ig ($L''$-ig) a kiértékelt tagok száma azon feltétel mellett, hogy $L' = L'' = i$

$\xi_3$: $L(\overline{L})$ esetén $L'$-ig ($L''$-ig) a kiértékelt tagok száma azon feltétel mellett, hogy $L'' = i$

Nyilván $\rho(L) \leqslant \rho(\overline{L})$ akkor és csak akkor igaz, ha

$$E(\xi_1 + a)p_1 + E(\xi_2 + a)p_2 + E(\xi_3 + \xi + b)p_3 \leqslant$$

$$\leqslant E(\xi_1 + \eta + a)p_1 + E(\xi_2 + b)p_2 + E(\xi_3 + b)p_3 .$$

Ennek átrendezésével adódik (2).

Megjegyzések

a.) A modell konkrét megvalosulásának leirása [1]-ben, [5]-ben és [3]-ban megtalálható. [4]-ben a problémát csak épp, hogy megemlitjük, jelen cikk az ott leirtak továbbfejlesztését adja. [6]-ban példán keresztül világitjuk meg a probléma lényegét.

b.) A minimalizálás programmal történő végrehajtása nem volt célunk, csupán a probléma feltevése és matematikai megoldása. A szükséges program megirása – a felvetődő igényeknek megfelelően – az [1]-ben leirt rendszer hatékonyságát fogja növelni.

c.) A forditóprogramok Boole-kifejezések optimalizálására való törekvése, illetve esetleges megvalósitása a logikai kifejezés 2. pontban leirt felbontását feleslegessé teheti, de a 3. pontban emlitett tételek ekkor is lényegesek.

d.) Eredményeink döntési eljárások (pl. döntési táblázatok) használatánál is alkalmazhatók.

## I r o d a l o m

[1] Korházi morbiditási adatok feldolgozását szolgáló statisztikai programrendszer (MTA SzTAKI, témadokumentáció)

[2] Ruzsa Imre – Urbán János: Matematikai logika (Tankönykiadó, 1969)

[3] Ruda Mihály: Egy széles körben alkalmazható programoptimalizálási módszer (MTA SzTAKI, Közlemények 20.)

[4] Krámli András – Ratkó István – Ruda Mihály – Soltész János: A sttisztikai adatfeldolgozás matematikai és számitástechnikai problémái, MTA SzTAKI, Tanulmányok (70/1977)

[5] Ratkó István: Egy számitástechnikai eszköz bonyolult logikai kifejezések leirására orvosstatisztikai alkalmazásokban (Számitástechnikai és kibernetikai módszerek alkalmazása az orvostudományban és a biólogiában. 3. Kollokvium, Szeged, 1977.)

[6] Halassy Béla – Zentai Tamás: Döntési táblázatok (NSZÁMOK, 1973.)

## РЕЗЮМЕ

### Проблемы оптимализации логических выражений в языке программирования

Иштван Ратко

Пусть $L_k$ состоит из $\ell_k$ конъюкций в логическом выражении

$$L = L_1 v\ L_2 v \dots v\ L_d$$

Число

$$\rho(L) = \ell_1 + \ell_2 + \dots + \ell_k + a_{j+1}$$

называется <u>стоимостью</u> L, если $\ell_j$-ая конъюкция в $L_j$ ложна, 1-ая,...,($\ell_j$-1)-ая конъюнкция в $L_j$ верна; если $L_1$ верна, пусть $\rho(L) = a_1$.

Если порядок дисюнкций и конъюнкций в дисюнкции изменяется, тогда $\rho(L)$ тоже изменяется.

В данной работе автор занимается минимализацией $\rho(L)$ и практической задачей выдвинула проблему.

## S u m m a r y

### On optimization problems of logical expressions in programming languages

István Ratkó

In the logical expression

$$L = L_1 \vee L_2 \vee \ldots \vee L_d$$

let $L_k$ consist of the conjuctions $I_k$ $(k = 1,2, \ldots, d)$.
We call the number

$$\rho(L) = I_1 + I_2 + \ldots + I_j + a_{j+1}$$

evaluating number of $L$ iff the $I_j$-th conjuction of is false, 1-st, . . . , $(I_j$-1)-st conjuctions of $L_j$ are true and the number of the conjuction signs in $L_{j+1}$ is $a_{j+1}$; if $L_1$ is true let $\rho(L)$ be $a_1$.

If we make a change in the order of the disjunctions and that of the conjunctions of its members, $\rho(L)$ changes, too.

In the paper we deal with minimizing $\rho(L)$ and with a partical problem which led to ours.

MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények** 20/1978.

# AZ "OPTIMALIZÁLÁS MINMÁLIS INFORMÁCIÓVAL" MÓDSZER ALLOKÁCIÓS FELADATOKRA VONATKOZÓ ALKALMAZÁSÁNAK EGY ÁLTALÁNOSITÁSA

Farkas Zoltán

## Bevezetés

A gyakorlat által felvetett optimális szétosztási (allokációs) problémák jelentős részének matematikai modellje olyan tipusu szélsőértékfeladat formájában fogalmazható meg, amelyben a rendelkezésre álló információtartalom nem elegendő konkrét célfüggvény megkonstruálásához. Az ilyen esetekben használatos közelitő megoldási módszerek helyett itt egy olyan más jellegű módszer egy általánositott változatát alkalmazzuk, amelyet egy más tipusu feladatra és speciálisabb esetre DOBÓ ANDOR és SZAJCZ SÁNDOR [1] cikkében vetett fel, és [2] -ben használta rá az "optimalizálás minimális információval" elnevezést, majd szerző [3], [4], [5]-ben kiterjesztette allokációs feladatokra is.

A közlendő algoritmusok az emlitetteknél általánosabb tipusu allokációs feladatok megoldására alkalmazhatók. Az irodalomban elég részletesen és általános formában tárgyalt ilyen problématipusok mindegyike feltételezi a célfüggvény konkrét alakjának ismeretét (ld. pl. R. BELLMAN [7] könyvének tárgyalásmódját), sőt igen erős feltételek – pl. differenciálhatóság, stb. – teljesülését (vö.: W. KARUSH [8] és T.A. SAATY – J. BRAM [9]), itt viszont ilyen jellegü feltételekkel nem élünk.

A következőkben módszert és ehhez kapcsolódó megoldási algoritmust is adunk jól meghatározott – de igen általános – tulajdonságokkal rendelkező, egyébként konkrét alakjában ismeretlen célfüggvények – bizonyos tipusu – feltételes szélsőértékhelyeinek meghatározására. Könnyen programozható algoritmus megadása a célunk.

Az ilyen megoldás legfőbb újszerüsége és legnagyobb előnye tehát az, hogy módszert és eljárást ad egy teljes függvényosztály elemei feltételes szélsőértékhelyeinek meghatározására.

Bevezetőben lássunk egy olyan allokációs probémát, amelynek kapcsán kialakitunk majd egy matematikai modellt, és ebben a felvetett problémára is érvényes megoldást (algoritmust) adunk meg.

Tételezzük fel, hogy egy termelő berendezés – mint erőforrás – $m$ számu különféle terméket állit elő $\underline{a} \epsilon R^m$ szintü összmennyiségben, és ezekből $n$ számu fogyasztónak szállit rendre $\underline{a}_1, \ldots, \underline{a}_i, \ldots, \underline{a}_n$ szintü mennyiségeket

$$(\underline{a} = \sum_{i=1}^{n} \underline{a}_i) \qquad \text{ahol az} \qquad \underline{a}_i = (a_{i1}, \ldots, a_{im}) \epsilon R^m \qquad i = 1, \ldots, n$$

vektorokban valamely $a_{ij} (1 \leqslant i \leqslant n,\ 1 \leqslant j \leqslant m)$ komponens értéke a termelő berendezéstől az $i$-edik fogyasztónak a $j$-edik termékből szállitott mennyiség.

Gyakran igény merül fel egy-egy fogyasztó felhasználási szintjének a növelésére, miközben a többi felhasználási szint sem csökkenhet. Ennek teljesitése bizonyos beruházási ráforditást igényel, amelynek az egyes fogyasztók közötti elosztása – a termékmennyiség elosztásától függően – különböző lehet. A ráforditás mértéke – valamilyen egységben mérve – számos tényező függvénye lehet. Igy pl. függhet a fogyasztónak az erőforrástól való távolságától, a rendelkezésre álló munkaeszközök szinvonalától, stb. A ráforditások tehát a befolyásoló tényezők függvényében vizsgálhatók, azonban ezek annyira sokrétü hatást jelenthetnek, hogy konkrét függvénykapcsolatok megállapitása a legtöbb esetben nem lehetséges. Ezért más utat kell keresnünk a megoldást illetően.

A felmerülő problémák egy jelentős részénél szerencsére indokoltak és célszerünek látszik az a heurisztikus feltételezés, hogy egy $\underline{a}_1$ felhasználási szintnek valamely $\underline{b}_1 > \underline{a}_1$ szintre való növelése nem követelhet nagyobb ráforditást, mint egy $\underline{a}_2$-ről $\underline{b}_2$-re való növelés, ha itt $\underline{a}_1 < \underline{a}_2$, és még fennáll $\underline{b}_1 - \underline{a}_1 = \underline{b}_2 - \underline{a}_2$. Tehát a ráforditásokat kifejező függvénynek rendelkeznie kell azzal a tulajdonsággal, hogy nagyobb kapacitásszintnek ugyanannyival történő megnövelése nagyobb ráforditást igényel, mint egy alacsonyabb szintnek a megnövelése. Ez másképpen megfogalmazva azt jelenti, hogy egy tökéletesebb termelő berendezés fajlagos tökéletesitése nagyobb ráforditást igényel, mint egy kevésbé tökéletes berendezésé. Ez a tulajdonság fontos lesz a modellalkotás során, mert alapját képezi bizonyos lényegi összefüggések megállapitásának.

Felmerül természetesen az az alapvető kérdés, hogy egy $\underline{a}$ kapacitásszintnek egy $\underline{b} > \underline{a}$ szintre való növelése során az $\underline{a}_1, \ldots, \underline{a}_i, \ldots, \underline{a}_n$ szinteket külön-külön hogyan kell megnövelni – feltéve, hogy a változatlanul hagyást igen, a csökkentés viszont nem engedjük meg – rendre valamilyen

$$\underline{b}_1, \ldots, \underline{b}_i, \ldots, \underline{b}_n \text{ szintre } (\underline{b}_i \geqslant \underline{a}_i, \quad i = 1, \ldots, n, \quad \underline{b} = \sum_{i=1}^{n} \underline{b}_i)$$

ahhoz, hogy ezt minimális ráforditással érjük el. Ez – a célfüggvény konkrét ismerete hiányában – azon ”minimális információ”ismeretében kell megválaszolnunk, amelyet az imént vázlatosan leirtunk.

Emlitést érdemel, hogy az imént vázolt probléma egy speciális ( m forráshelyü, n felvevőhelyü), de – nemlineáris célfüggvénye miatt – bizonyos értelemben általánositott szállitási feladatként is interpretálható.

Megemlitjük még, hogy az [1] alatt hivatkozott cikkbeli megbizhatóságelméleti feladat egy általánositásához juthatunk, ha a termelő berendezések különböző paramétereik szerinti müködési megbizhatóságait akarjuk adott szintről megnövelni úgy, hogy az eredő megbizhatóság egy – az eredetinél nagyobb – adott vektor. A közlendő megoldási algoritmusból levezethető erre az általánosabb esetre vonatkozó megoldás is.

A szerző ezúton is szeretne köszönetet mondani [1] cikk szerzőinek korábbi problémafelvetéseikért, és szakmai támogatásukért.

**A problémakör matematikai leirása**

Jelölje $M(\underline{x},\underline{y})$ egy olyan ráforditás mértékét, amelyet akkor végzünk, ha a termelési volumen $\underline{x}$ szintjét $\underline{y}$-ra módositjuk úgy, hogy az $\underline{x}$ vektorral jelölt szint egyetlen komponensét sem csökkentjük ($\underline{0} \leqslant \underline{x} \leqslant \underline{y} \leqslant \underline{\omega}$, $\underline{0},\underline{\omega},\underline{x},\underline{y} \in R^m$, $\underline{\omega}$ adott kapacitáskorlát ).

A továbbiakban az $M(\underline{x},\underline{y})$ függvényt nevezzük munka- vagy beruházási ráforditásfüggvénynek, vagy röviden csak *ráforditásfüggvénynek,* ha minden olyan $\underline{x},\underline{y},\underline{z}$ vektorok esetén teljesülnek rá az

$1^0 \quad 0 < M(\underline{x},\underline{y}) < \infty \qquad (\underline{0} \leqslant \underline{x} < \underline{y} < \underline{\omega})$

$2^0 \quad M(\underline{x},\underline{y}) + M(\underline{y},\underline{z}) = M(\underline{x},\underline{z}),$

feltételek, amelyekre $\underline{0} \leqslant \underline{x} \leqslant \underline{y} \leqslant \underline{z} < \underline{\omega}$ és $\underline{0},\underline{\omega},\underline{x},\underline{y},\underline{z} \in R^m$ ($\underline{\omega}$ adott)

**Megjegyzések**

1. Nyilvánvaló, hogy az $M(\underline{x},\underline{y})$ függvény csak azon

$$(\underline{x},\underline{y}) \in (X^m_{i=1}[0,\omega_i]) \times (X^m_{i=1}[0,\omega_i])$$

rendezett párokra van értelmezve, amelyekre az $\underline{x} \leqslant \underline{y}$ rendezési reláció teljesül, és hogy ez $m = 1$ esetén mindig fennáll. Ilyen vonatkozásban is általánositása ez az [1]-beli modellnek, amely az itteni vektorváltozók helyett skalárváltozókat használ.

2. Az $M(\underline{x},\underline{y})$ függvényre $\underline{x} < \underline{y} < \underline{z}$ esetén az alábbi tulajdonságok könnyen beláthatók:

1. $M(\underline{x},\underline{y}) < M(\underline{x},\underline{z})$
2. $M(\underline{x},\underline{z}) > M(\underline{y},z)$
3. $M(\underline{x},\underline{x}) = 0$
4. $M(\underline{x},\underline{y}) = M(\underline{0},\underline{y}) - M(\underline{0},\underline{x})$

Az $M(\underline{0},x) = H(\underline{x})$ jelölés bevezetésével a 4. tulajdonság egyszerübben igy irható fel:

$4' \; M(\underline{x},\underline{y}) = H(\underline{y}) - H(\underline{x}),$

tehát a $H(\underline{x})$ függvény monoton növekvő, és $H(\underline{0}) = 0$.

A $2^0$ feltételből ugyanis az $1^0$-et is figyelembevéve:

$$0 < M(\underline{y},\underline{z}) = M(\underline{x},\underline{z}) - M(\underline{x},\underline{y})$$

és

$$0 < M(\underline{x},\underline{y}) = M(\underline{x},\underline{z}) - M(\underline{y},\underline{z}),$$

ami ekvivalens az 1. és 2. tulajdonsággal.

Az $\underline{x} = \underline{y} = \underline{z}$ esetben a $2^0$ feltétel a 3. tulajdonságot adja. Végül az $\underline{x} = \underline{0}$ választással, $\underline{y} = \underline{x}$ és $\underline{z} = \underline{y}$ helyettesitéssel a $2^0$ feltétel éppen 4. alakban irható. Ennek az állitásnak a megforditottja is igaz, amelyet lemma formájában mondunk ki.

**1. Lemma.** *Ha valamely $M(\underline{x},\underline{y})$ függvény rendelkezik az 1.–4. tulajdonságokkal, akkor kielégiti az $1^0$ és $2^0$ feltételeket is. (Bizonyitás az [1]-beli analógiájára végezhető el, skalárok helyett vektorokkal.)*

**2. Lemma.** *Tegyük fel, hogy teljesülnek az $M(\underline{x},\underline{y})$ függvényre az*

$$1^0 \quad 0 < M(\underline{x},\underline{y}) < \infty, \quad \underline{0} < \underline{x} < \underline{y} < \underline{\omega}$$

$$2^0 \quad M(\underline{x},\underline{y}) + M(\underline{y},z) = M(\underline{x},\underline{z}), \underline{0} \leqslant \underline{x} \leqslant \underline{y} \leqslant \underline{z} < \underline{\omega}.$$

*összefüggések, ahol $\underline{0},\underline{x},\underline{y},\underline{z},\underline{\omega} \in R^m$.*
*Annak szükséges és elégséges feltétele, hogy a*

$$3^0 \quad M(\underline{x},\underline{x} + \underline{\delta}) \leqslant M(\underline{y},\underline{y} + \underline{\delta})$$

*egyenlőtlenség fennálljon az, hogy*

$$3^{0'} \quad M(\underline{x},\underline{y}) \leqslant M(\underline{x} + \underline{\delta}, \underline{y} + \underline{\delta})$$

*egyenlőtlenség teljesüljön a $\underline{0} \leqslant \underline{x} < \underline{y} < \underline{\omega}$ és $\underline{\delta} > \underline{0}$ (de $\underline{y} + \underline{\delta} < \underline{\omega}$) vektorok esetén.*

**Bizonyitás.** Közvetlenül az értelmezésből következik, hiszen

$$3^0 \quad H(\underline{x} + \underline{\delta}) - H(\underline{x}) \leqslant H(\underline{y} + \underline{\delta}) - H(\underline{y})$$

akkor és csak akkor teljesül, ha

$$3^0 \quad H(\underline{y}) - H(\underline{x}) \leqslant H(\underline{y} + \underline{\delta}) - H(\underline{x} + \underline{\delta})$$

**3. Lemma.** *A $1^0$ és $2^0$ feltételeknek eleget tevő függvényekre teljesüljön meg az alábbi feltétel:*

$$3^0 \quad M(\underline{x},\underline{x} + \underline{\delta}) \leqslant M(\underline{y},\underline{y} + \underline{\delta})$$

*minden $\underline{0} \leqslant \underline{x} < \underline{y} < \underline{\omega}$ és tetszöleges $\underline{\delta} : \underline{0} < \underline{\delta} < \underline{y} + \underline{\delta} < \underline{\omega}$ esetén.*

*Ekkor $H(\underline{x})$ konvex függvény és $3^0$-ban egyenlőség akkor és csak akkor áll fenn, ha $H(\underline{x})$ egyúttal lineáris is.*

**Megjegyzés.**

A $3^0$ feltétel az [1] cikkbeli analóg feltétellel ellentétben megengedi az egyenlőséget is, tehát az ottaninál ilyen vonatkozásban is általánosabb feltételt fogalmaz meg.

**Bizonyitás.**

Az előző megjegyzésben foglaltak egyuttal azt is jelentik, hogy a bizonyitás is bővül az [1]-belihez képest. Bármely $\underline{\delta} \geqslant \underline{o}$ mellett, tetszőleges $\underline{x},\underline{y} : \underline{x} < \underline{y}$ esetén teljesül a $H(\underline{x} + \underline{\delta}) - H(\underline{x}) \leqslant H(\underline{y} + \underline{\delta}) - H(\underline{y})$ egyenlőtlenség, igy speciálisan az $\underline{x} + \underline{\delta} = \underline{y}$ feltételnek eleget tevő értékekre is teljesülnie kell.

Ez utóbbi összefüggést viszont tetszőleges $\underline{x}_1 < \underline{x}_2$ esetén kielégitik az $\underline{x} = \underline{x}_1, \underline{y} = (\underline{x}_1 + \underline{x}_2)/2, \underline{\delta} = (\underline{x}_2 - \underline{x}_1)/2$ egyenlőségekkel definiált értékek. Tehát igy fennáll, hogy

$$H(\underline{x}_1 + \underline{\delta}) - H(\underline{x}_1) \leqslant H(\frac{\underline{x}_1 + \underline{x}_2}{2} + \delta) - H(\frac{\underline{x}_1 + \underline{x}_2}{2})$$

vagyis

$$H(\frac{\underline{x}_1 + \underline{x}_2}{2}) \leqslant \frac{H(\underline{x}_1) + H(\underline{x}_2)}{2}$$

Ez viszont éppen a konvexitás definiciója. Egyenlőség nyilván akkor és csak akkor áll, ha $3^0$-ban is az áll.

Még hátra van annak belátása, hogy ez utóbbi akkor és csak akkor teljesül tetszőleges $\underline{x}_1, \underline{x}_2$ vektorokra, ha $H(\underline{x})$ lineáris függvény.

Mielőtt rátérnénk a bizonyitásra megállapithatjuk, hogy $H(\underline{x})$ folytonos függvény. Az analizisből jól ismert tény ugyanis, hogy minden konvex függvény folytonos.
(Ebből már következik, hogy $M(\underline{x},\underline{y})$ is folytonos.)

Hogy lineáris $H(\underline{x})$ estén egyenlőség áll, az trivialitás. Forditott irányu állitásunk igazolásához elég bizonyitanunk, hogy az $M(\underline{x},\underline{x} + \underline{\delta}) = M(\underline{y},\underline{y} + \underline{\delta})$, vagy ami ugyanaz, a

$$H(\frac{\underline{x}_1 + \underline{x}_2}{2}) = \frac{H(\underline{x}_1) + H(\underline{x}_2)}{2}$$

függvényegyenletet kialégitő $H(\underline{x})$ függvények megoldásai eleget tesznek a

$$H(\underline{x}_1 + \underline{x}_2) = H(\underline{x}_1) + H(\underline{x}_2)$$

ismert Cauchy-tipusu függvényegyenletnek. Ugyanis $H(\underline{x})$ folytonos lévén ez utóbbiak – mint ismeretes – csak $H(\underline{x}) = c \cdot \underline{x}$ alaku lineáris megoldása lehetséges. (V.ö.: ACZÉL JÁNOS [6])

Viszont a feltételezés szerint:

$$H(\underline{x}_1) + H(\underline{x}_2) = 2 \cdot H(\frac{\underline{x}_1 + \underline{x}_2}{2}) = 2 \cdot H(\frac{(\underline{x}_1 + \underline{x}_2) + \underline{0}}{2}) =$$

$$= 2 \cdot \frac{H(\underline{x}_1 + \underline{x}_2) + H(\underline{0})}{2} = H(\underline{x}_1 + \underline{x}_2),$$

amivel állitásunkat igazoltuk.

**Megjegyzés.**

Ha egy függvény konvex, akkor létezik legalább egyoldali deriváltja is. Deriválhatóság azonban nyilvánvalóan nem következik a konvexitásból, mert pl. törtvonalak által meghatározott konvex függvény töréspontjaiban csakis egyoldali derivált létezik.

A továbbiakban nem használjuk fel a $H(\underline{x})$ függvényeknek még ezen utóbbi tulajdonságát sem, pusztán a III. Lemmában bizonyitottakat.

**Elnevezés.**

Az $1^0$, $2^0$, $3^0$ feltételeknek eleget tevő $M(\underline{x},\underline{y})$ függvényeket nevezzük megengedett (munka-, vagy beruházási) ráforditásfüggvénynek.

**Megjegyzés.**

Minden monoton nemcsökkenő konvex $H(\underline{x})$ függvényből megengedett ráforditásfüggvény adódik az $M(\underline{x},\underline{y}) = H(\underline{y}) - H(\underline{x})$ definició révén.

A bevezetőben példaként leirt allokációs problémák kapcsán a megengedett ráforditásfüggvényekre általánosan megformulázható problémakör: keressük a

$$\sum_{i=1}^{n} M_i(\underline{x}_i,\underline{y}_i) \qquad (\text{ahol } M_i(\underline{x}_i\underline{y}_i) = M(\underline{x}_i\underline{y}_i), \qquad i = 1,2,\ldots,n)$$

tipusu célfüggvények feltételes minimumhelyeit az

$$(S) \qquad \sum_{i=1}^{n} \underline{y}_i = \underline{y} \qquad (\text{vagy } \prod_{i=1}^{n} y_i^{(j)} = y^{(j)}, j = 1,\ldots,m)$$

tipusu feltétel mellett, ahol $\underline{0} \leqslant \underline{x}_i < \underline{\omega}$, $\underline{0} < \underline{y}_i < \underline{\omega}$, $\underline{x}_i \leqslant \underline{y}_i$ ($\underline{\omega}$ adott), valamint $M_i(\underline{x}_i\,\underline{y}_i)$ megengedett ráforditásfüggvény minden $i = 1,2,\ldots,n$ esetén,

$$\underline{y} \quad (\underline{y} \geqslant \sum_{i=1}^{n} \underline{x}_i, \text{ vagy } [y_i^{(j)} > \prod^{n} x_i^{(j)}]_{j=1}^{m}) \quad .$$

pedig megadott konstans vektor, az un. kapacitásvektor. Mint a későbbiek során fogjuk látni, a fenti "szorzat-tipusú" feltétellel megfogalmazható feladat visszavezethető az "összeg-tipusú" feltételes feladat megoldására.

**A probléma megoldása, optimalizálási algoritmusok**

Abban az esetben, ha a megengedett ráforditásfüggvény definiálásakor az $\underline{x}_i$ és $\underline{y}_i$ közt semmiféle egyenlőtlenségi reláció teljesülését nem követelnénk meg, akkor a probléma megoldása leegyszerüsödne. (A bevezetőben emlitett gyakorlati problémára vonatkozóan is lényeges kikötés volt az $\underline{a}_i \leqslant \underline{b}_i$ feltétel teljesülése!)

Különösen fennáll ez, ha az egyébként is szükségessé váló $M_i(\underline{x}_i,\underline{y}_i) = M(\underline{x}_i,\underline{y}_i)$ feltevéssel élünk ($i = 1,2,\ldots,n$). Ekkor ugyanis a konvex függvényekre érvényes Jensen-egyenlőtlenség folytán az következne, hogy

$$(2) \qquad n \cdot H\left(\frac{\sum_{i=1}^{n} \underline{y}_i}{n}\right) \leqslant \sum_{i=1}^{n} H(\underline{y}_i)$$

Figyelembe véve azt, hogy $H(\underline{x})$ monoton növekvő függvény, a baloldalnak ott van minimuma, ahol a

$$\sum_{i=1}^{n} \underline{y}_i$$

kifejezésnek. Az (s) tipusu feltétel mellett ez konstans, és igy az

$$(3) \qquad n \cdot H\left(\frac{\underline{y}}{n}\right) \leqslant \sum_{i=1}^{n} H(\underline{y}_i)$$

egyenlőtlenség miatt az $\underline{y}_i = \frac{\underline{y}}{n}$ minimumhelyet ad ($i = 1,2,\ldots,n$). Az $\underline{x}_i \leqslant \underline{y}_i (i = 1,2,\ldots,n)$ reláció előirása esetén azonban ennél bonyolultabb megoldó algoritmussal lehet csak meghatározni a minimumhelyet. Erre vonatkozik a következő tétel.

**Tétel.**

*Legyen* $M_i(\underline{x}_i,\underline{y}_i) = M(\underline{x}_i,\underline{y}_i)$ $i = 1,2,\ldots,n$ *megengedett ráforditásfüggvény (tehát az* $\underline{x}_i \leqslant \underline{y}_i$ *egyenlőtlenség teljesüljön minden* $i = 1,2,\ldots,n$ *eseten). Legyenek továbbá az* $\underline{x}_1,\underline{x}_2,\ldots,\underline{x}_n \in R^m$ $(\underline{0} \leqslant \underline{x}_1 \leqslant \ldots \leqslant \underline{x}_n < \underline{\omega})$, *valamint az* $\underline{y} \in R^m$ $(\underline{y} > \sum_{i=1}^{n} \underline{x}_i)$ *vektorok rögzitve. A*

$$(4) \qquad \sum_{i=1}^{n} M(\underline{x}_i, \underline{x}_i + \delta_i)$$

*függvénynek a* $[\underline{0},\underline{\omega}]$ *intervallumban a* $\underline{\delta}_1 \geqslant \underline{0}$, $i = 1,2,\ldots,n$, továbbá az

$$(5) \qquad \sum_{i=1}^{n} (\underline{x}_1 + \underline{\delta}_1) = \underline{y}$$

*feltétel teljesülése esetén felveszi minimumát, és minimuhelyeket az* $M(\underline{x},\underline{y})$ *függvény konkrét alakjától függetlenül meghatározhatunk, pl. a következő algoritmussal:*

*Algoritmus*: a feltételeket teljesitő (4) célfüggvény minimumát a

$$(6)\qquad \underline{\delta}_i^* = \begin{cases} \frac{1}{j}(\underline{y} - \sum_{\lambda=j+1}^{n} \underline{x}_\lambda) - \underline{x}_1, & \text{ha} \quad i = 1,2,\ldots,j \\ \underline{0} & \text{ha} \quad i = j+1,\ldots,n \end{cases}$$

vektorrendszeren biztosan felveszi, feltéve, hogy létezik olyan $j$ küszöbindex, amellyel a

$$(7)\qquad \underline{s}_j < \underline{y} \leqslant \underline{s}_{j+1}$$

egyenlőtlenség teljesül. A legkisebb ilyen $j$ értéket a következő vektorrendszerből határozhatjuk meg:

$$(8)\qquad \begin{array}{l} \underline{x}_1 + \underline{x}_2 + \ldots + \underline{x}_n = \underline{s}_1 \\ \quad 2\underline{x}_2 + \underline{x}_3 + \ldots + \underline{x}_n = \underline{s}_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \quad\quad j \cdot \underline{x}_j + \underline{x}_{j+1} + \ldots + \underline{x}_n = \underline{s}_j \\ \quad\quad (j+1)\underline{x}_{j+1} + \ldots + \underline{x}_n = \underline{s}_{j+1} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \quad\quad\quad n\underline{x}_n = \underline{s}_n \\ \quad\quad\quad (n+1)\underline{y} = \underline{s}_{n+1} \end{array}$$

Ha a $H(\underline{x}) = M(\underline{0},\underline{x})$ függvény mindenütt szigorúan konvex (vagy ami esetünkben ugyanaz: sehol sem lineáris), akkor (6) a tételbeli minimumfeladat egyetlen megoldása.

**Bizonyitás.**

Előzetesen megemlitjük, hogy a (8) egyenletek a következő rekurziv formában is megadhatók:

$$\underline{s}_1 = \sum_{i=1}^{n} \underline{x}_i$$

$$(8')\qquad \underline{s}_{i+1} = \begin{cases} (\underline{x}_{i+1} - \underline{x}_i)i + \underline{s}_i, & \text{ha} \quad i = 1,2,\ldots,n-1 \\ (n+1)\underline{y}, & \text{ha} \quad i = n \end{cases}$$

Ebből viszont közvetlenül következik, hogy

$$\underline{s}_1 \leq \underline{s}_2 \leq \ldots \leq \underline{s}_j \leq \underline{s}_{j+1} \leq \ldots \leq \underline{s}_n \leq \underline{s}_{n+1}$$

Vagyis az, hogy a (7) egyenlőtlenség annak (8)-beli értelmezésével együtt valóban egy és csak egy olyan $j$ küszöbindexet definiál, ameddig bezárólag az $x_i$ vektorokat meg kell növelni, és a hátralévőket változatlanul kell hagyni a tételben optimálisnak mondott megoldás előállitásához. Erre a megoldásra tehát teljesül:

$$(9) \qquad \frac{1}{j}(\underline{y} - \sum_{\lambda=j+1}^{n} \underline{x}_\lambda) = \underline{x}_1 + \underline{\delta}_1^* = \ldots = \underline{x}_j + \underline{\delta}_j^* = \underline{z},$$

amelyre $j < n$ esetén

$$\underline{z} \leq \underline{x}_{j+1} \leq \ldots \leq \underline{x}_n$$

$j = n$ esetén pedig $\underline{y}/n = \underline{z} > \underline{x}_n$

Igen könnyen ellenőrizhető, hogy ez megengedett ( a feltételrendszert kielégitő) megoldás. Az is nyilvánvaló, hogy a (4) célfüggvénynek ugyanott van minimuma, mint a

$H(\underline{x}_i + \underline{\delta}_i)$ függvénynek, továbbá, hogy utóbbinak a $[\underline{0},\underline{\omega}]$ zárt intervallumban az adott feltételek mellett létezik minimuma.

A tétel most következő bizonyitása során azt a két alapvető esetet különböztessük meg, hogy $j < n$, vagy $j = n$.

Kezdjük a $j = n$ esettel. Ekkor közvetlenül alkalmazható a Jensen-egyenlőtlenség, hiszen az $\underline{x}_i$ vektorok mindegyike megnövelve szerepel a (6) alatt megadott vektorrendszerben.

Definiálja tehát a $\underline{\delta}_i$ $i = 1, \ldots, n$ vektorrendszer a minimumfeladat tetszőleges megengedett megoldását. Ekkor

$$\sum_{i=1}^{n} H(\underline{x}_i + \underline{\delta}_i^*) = n \cdot H\left(\frac{\sum_{i=1}^{n}(\underline{x}_i + \underline{\delta}_i^*)}{n}\right) = nH\left(\frac{\underline{y}}{n}\right) =$$

$$(10) \qquad = n \cdot H\left(\frac{\sum_{i=1}^{n}(\underline{x}_i + \underline{\delta}_i)}{n}\right) \leq$$

$$\leq \sum_{i=1}^{n} H(\underline{x}_i + \underline{\delta}_i)$$

vagyis (6) valóban optimális megoldást szolgáltat.

Ha a $\{\underline{\delta}_i\}$ vektorrendszer nem azonos $\{\underline{\delta}_i^*\}$-gal ($i = 1, \ldots, n$), és a konvexitás (vagyis a $3^0$ feltételi egyenlőtlenség) szigorú formájában teljesül, akkor a (10)-beli egyenlőtlenség is szigoru formában teljesül, vagyis (6) az egyedüli megoldás.

Most térjünk át a $j < n$ esetbeli bizonyitásra, amelyben az előző eset bizonyitásával ellentétben teljes indukciót használunk. Itt nyilván csak az $n \geqslant 2$ eset jön számitásba. $n = 2$ esetén (ha tehát $\underline{s}_1 < \underline{y} < \underline{s}_2$) pedig a (6) vektorrendszer a következő tulajdonságú:

$$\underline{y} - \underline{x}_2 = \underline{x}_1 + \underline{\delta}_1^*$$

Vagyis az $\underline{x}_1 + \underline{\delta}_1^*$ vektort olyan $\underline{\Delta}$-val csökkentve, hogy $\underline{x}_1 + \underline{\delta}_1^* - \underline{\Delta} \geqslant \underline{x}_1$ maradjon, és ezzel a $\underline{\Delta}$-val növelve az $\underline{x}_2$ vektort (tehát egy tetszőleges megengedett megoldást előállitva), az $M(\underline{x})$ függvény $3^0$ tulajdonsága miatt a

$$H(\underline{x}_1 + \underline{\delta}_1^*) + H(\underline{x}_2) \leqslant H(\underline{x}_1 + \underline{\delta}_1^* - \underline{\Delta}) + H(\underline{x}_2 + \underline{\Delta})$$

egyenlőtlenség adódik. Tehát a (6) algoritmus $n = 2$ esetén optimális megoldást ad, és $H(\underline{x})$ szigorú konvexitása (vagyis a $3^0$-ban a szigorú egyenlőtlenség mindenütt való teljesülése) esetén ez az egyedüli megoldás.

Most tételezzük fel, hogy tetszőleges $n = k$ esetén a (6) vektorrendszer valóban optimális megoldást szolgáltat, és bizonyitsuk ezt be ezt $n = k + 1$ esetére is.

Legyen $\underline{y} = \underline{y}^{(n)}$, és jelölje a (6) algoritmus révén – az $n$ paraméter függvényében tekintve – felirt megengedett megoldást az

$$(11) \qquad \underline{x}_i + \underline{\delta}_i^{(n)} \qquad i = 1, 2, \ldots, n$$

vektorrendszer, azon küszöbindexet pedig, ameddig bezárólag az $\underline{x}_i$ vektorokat meg kell növelni: $j_n$. Ekkor az $n = k + 1$ esetben ($j_{k+1} < k + 1$ feltételezése miatt):

$$(12) \qquad \frac{1}{j_{k+1}} \left(\underline{y}^{(k+1)} - \sum_{\lambda = j_{k+1}+1}^{k} \underline{x}_\lambda\right) = \underline{x}_1 + \underline{\delta}_1^{(k+1)} = \underline{x}_2 + \underline{\delta}_2^{(k+1)} = \ldots = \underline{x}_{j_{k+1}}' + \underline{\delta}_{j_{k+1}}^{(k)} \leqslant$$

$$\leqslant \underline{x}_{j_{k+1}+1} \leqslant \ldots \leqslant \underline{x}_{k+1},$$

vagyis $\underline{\delta}_{k+1}^{(k+1)} = \underline{0}$ biztosan teljesül. Azt kell belátnunk, hogy a (12) vektorrendszer optimális megoldást szolgáltat.

Az $j = n$ esetére adott korábbi bizonyitás és az első indukciós feltevés alapján az a tény könnyen belátható, hogy az $\underline{x}_{k+1}$ vektor elhagyásával keletkező $\underline{y}^{(k)} = \underline{y}^{(k+1)} - \underline{x}_{k+1}$ kapacitásvektorral definiálható feladat (amikoris $n = k$) (6) vektorrendszer által meghatározott optimális megoldása megegyezik a (11) megengedett megoldás első $k$ komponensével (ami egyuttal azt is jelenti, hogy a (6) algoritmus által a két szóbanforgó feladatra meghatározott küszöbindex megegyezik: $j_{k+1} = j_k$), tehát

$$\underline{x}_i^{(k+1)} + \underline{\delta}_i^{(k+1)} = \underline{x}_i^{(k)} + \underline{\delta}_i^{(k)}, \quad \text{ha} \quad i = 1,2,\ldots,k$$

Térjünk most át a célfüggvény vizsgálatára. Jelölje $F_{k+1}(\underline{y}^{(k+1)})$, illetve $F_k^*(\underline{y}^{(k)} - \underline{x}_{k+1})$ és $f_k(\underline{y}^{(k)} - \underline{x}_{k+1})$ rendre az $n = k + 1$, illetve $n = k$ esetbeli feladat egy – a (6) vektorrendszer által meghatározott – megengedett, illetve optimális és tetszőleges megengedett megoldáshoz tartozó célfüggvényértékeket (az argumentumok rendre a megfelelő kapacitásvektorok). Ekkor fennáll a következő egyenlőtlenség:

$$\begin{aligned} F_{k+1}(\underline{y}^{(k+1)}) &= \sum_{i=1}^{k+1} H(x_i + \underline{\delta}_i^{(k+1)}) = \\ &= \sum_{i=1}^{k} H(\underline{x}_i + \underline{\delta}_i^{(k)}) + H(\underline{x}_{k+1}) = \\ &= F_k^*(\underline{y}^{(k+1)} - \underline{x}_{k+1}) + H(\underline{x}_{k+1}) \leqslant \\ &\leqslant f_k(\underline{y}^{(k+1)} - \underline{x}_{k+1}) + H(\underline{x}_{k+1}) = f_{k+1}(\underline{y}^{(k+1)}) \end{aligned} \tag{13}$$

Eszerint tehát a (6) algoritmus tetszőleges olyan megengedett megoldások közül optimális megoldást ad $n = k + 1$ esetén, amelyben $\underline{\delta}_{k+1}^{(k+1)} = \underline{0}$.

Most bebizonyitjuk, hogy azon megengedett megoldások közül, amelyekben $\underline{\delta}_{k+1} > \underline{0}$, mindegyik nem kisebb (szigorú konvexitása esetén határozottan nagyobb) célfüggvényértékét szolgáltat a (6) vektorrendszer által meghatározottnál.

Legyen ugyanis egy ilyen tulajdonságú, de egyébként tetszőlegesen megengedett vektorrendszer (mint megengedett megoldás) a következő:

$$\underline{x}_1 + \underline{\delta}_1, \ \underline{x}_2 + \underline{\delta}_2, \ldots, \underline{x}_{k+1} + \underline{\delta}_{k+1},$$

ahol

$$\sum_{i=1}^{k} (\underline{x}_i + \underline{\delta}_i) = \underline{y}^{(k+1)} - \underline{x}_{k+1} - \underline{\delta}_{k+1}, \quad \text{és} \quad \underline{\delta}_{k+1} > \underline{0}$$

A célfüggvényre áttérve, használjuk a korábbi jelöléseket:

$$f_{k+1}(\underline{y}^{(k+1)}) = f_k(\underline{y}^{(k+1)} - \underline{x}_{k+1} - \underline{\delta}_{k+1}) + H(\underline{x}_{k+1} + \underline{\delta}_{k+1}) \overset{\text{def}}{=}$$

$$= \sum_{i=1}^{k} H(\underline{x}_i + \underline{\delta}_i) + H(\underline{x}_{k+1} + \underline{\delta}_{k+1}) \geqslant \text{(indukciós feltevés)}$$

$$\geqslant F_k^*(\underline{y}^{(k+1)} - \underline{x}_{k+1} - \underline{\delta}_{k+1}) + H(\underline{x}_{k+1} + \underline{\delta}_{k+1}) \overset{\text{def}}{=}$$

$$(14) \qquad = \sum_{i=1}^{j_k} H(\underline{x}_i + \underline{\delta}_i^{(k)}) + \sum_{i=j_k+1}^{k} H(\underline{x}_i + \underline{\delta}_i^{(k)}) + H(\underline{x}_{k+1} + \underline{\delta}_{k+1}) \geqslant$$

$$\geqslant \sum_{i=1}^{j_k} H(\underline{x}_i + \underline{\delta}_i^{(k+1)}) + \sum_{i=j_k+1}^{j_{k+1}} H(\underline{x}_i + \underline{\delta}_i^{(k+1)}) + \sum_{i=j_{k+1}+1}^{k} H(\underline{x}_i) + H(\underline{x}_{k+1}) \overset{\text{def}}{=}$$

$$= F_k^*(\underline{y}^{(k+1)} - \underline{x}_{k+1}) + H(\underline{x}_{k+1}) \overset{\text{def}}{\geqslant} F_{k+1}^*(\underline{y}^{(k+1)})$$

A (14) egyenlőtlenség közül csak a második szorul bizonyitásra. Ez viszont pontosan akkor teljesül ha

$$\sum_{i=1}^{j_k} H(\underline{x}_i + \underline{\delta}_i^{(k)}) + \sum_{i=j_k+1}^{k} H(\underline{x}_i + \underline{\delta}_i^{(k)}) + H(\underline{x}_{k+1} + \underline{\delta}_{k+1}) \geqslant$$

$$\geqslant \sum_{i=1}^{j_k} H(\underline{x}_i + \underline{\delta}_i^{(k+1)}) + \sum_{i=j_k+1}^{j_{k+1}} H(\underline{x}_i + \underline{\delta}_i^{(k+1)}) + \sum_{i=j_{k+1}+1}^{k} H(\underline{x}_i) + H(\underline{x}_{k+1}).$$

Mivel $i > j_k$ esetén $\underline{\delta}_i^{(k)} = \underline{0}$, továbbá $i > j_{k+1}$ esetén $\underline{\delta}_i^{(k+1)} = \underline{0}$, ezért elegendő a következő egyenlőtlenséget igazolni:

$$(15) \qquad H(\underline{x}_{k+1} + \underline{\delta}_{k+1}) - H(\underline{x}_{k+1}) \geqslant \sum_{i=1}^{j_{k+1}} (H(\underline{x}_i + \underline{\delta}_i^{(k+1)}) - H(\underline{x}_i + \underline{\delta}_i^{(k)})).$$

Ehhez osszuk fel az $[\underline{x}_{k+1}, \underline{x}_{k+1} + \underline{\delta}_{k+1}]$ intervallumot rendre a

$$\underline{\delta}_1^{(k+1)} - \underline{\delta}_1^{(k)}, \underline{\delta}_2^{(k+1)} - \underline{\delta}_2^{(k)}, \ldots, \underline{\delta}_i^{(k+1)} - \underline{\delta}_i^{(k)}, \ldots, \underline{\delta}_{j_{k+1}}^{(k+1)} - \underline{\delta}_{j_{k+1}}^{(k)}$$

hosszuságú

$$[\underline{x}_{k+1} + \underline{\Delta}_0, \underline{x}_{k+1} + \underline{\Delta}_1], \ldots, [\underline{x}_{k+1} + \underline{\Delta}_{i-1}, \underline{x}_{k+1} + \underline{\Delta}_i], \ldots, [\underline{x}_{k+1} + \underline{\Delta}_{j_{k+1}-1}, \underline{x}_{k+1} + \underline{\Delta}_{j_{k+1}}]$$

intervallumokra, ahol a $\underline{\Delta}_i$ vektorokat a következő rekurziv összefüggéssel adhatjuk meg:

$$\underline{\Delta}_0 = \underline{0},\ \underline{\Delta}_i = \underline{\Delta}_{i-1} + (\underline{\delta}_i^{(k+1)} - \underline{\delta}_i^{(k)}), \qquad i = 1,2,\ldots,j_{k+1}$$

Ez a konstrukció lehetővé teszi a következő azonosság felirását:

$$(16) \qquad H(\underline{x}_{k+1} + \underline{\delta}_{k+1}) - H(\underline{x}_{k+1}) \equiv \sum_{i=1}^{j_{k+1}} (H(\underline{x}_{k+1} + \underline{\Delta}_i) - H(\underline{x}_{k+1} + \underline{\Delta}_i - (\underline{\delta}_i^{(k+1)} - \underline{\delta}_i^{(k)}))).$$

Mivel a $H$ függvény konvexitásából, valamint a $j_{k+1} < n$ feltevésből következik, hogy

$$(17) \qquad H(\underline{x}_i + \underline{\delta}_i^{(k+1)}) - H(\underline{x}_i + \underline{\delta}_i^{(k)}) \leqslant H(\underline{x}_{k+1} + \underline{\Delta}_i) - H(\underline{x}_i + \underline{\Delta}_i - (\underline{\delta}_i^{(k+1)} - \underline{\delta}_i^{(k)})),$$

ezért (16)-nak a (15)-be való helyettesitésekor tagonkénti (17) tipusu egyenlőtlenségeket konstatálhatunk. Ezáltal tehát igazoltuk magát a (15) egyenlőtlenséget is.

Szigorú konvexitás esetén (14)-ben a (17)-beli egyenlőtlenség szigoru értelemben való teljesülése miatt szigorú egyenlőtlenség áll, tehát a (6) vektorrendszer szolgáltatja az egyetlen optimális megoldást. Emiatt (11) valóban optimális megoldása a feladatnak, és szigorú konvexitása esetén az egyetlen. Ezzel a tétel állitását igazoltuk.

**Megjegyzések.**

1. Könnyen belátható, hogy a tételben közölt feltételeket teljesitő

$$\underline{y}_i = (\underline{y}_i^{(1)}, \underline{y}_i^{(2)}, \ldots, \underline{y}_i^{(m)})^T, \qquad i = 1,2,\ldots,n$$

vektorokra előirt

$$(\mathrm{P}) \qquad \prod_{i=1}^{n} y_i^{(j)} = y^{(j)} \qquad j = 1,2,\ldots,m$$

feltételrendszer mellett a (4) célfüggvényre megfogalmazott szélsőértékfeladat visszavezethető a tételben megfogalmazottra, és igy az [1]-beli megbizhatóságielméleti feladat egy általánositásának megoldását nyerhetjük. Ehhez elegendő a $z_i^{(j)} = \log y_i^{(j)} (j = 1,2,\ldots,m)$ uj változókat bevezetni, mert ekkor a (P) feltételrendszer a

$$\sum_{i=1}^{n} z_i^j = \log y^{(j)} \qquad j = 1,2,\ldots,m.$$

(S) tipusba megy át, és a célfüggvény igy módosul:

$$\sum_{i=1}^{n} H(\underline{y}_i) = \sum_{i=1}^{n} H((e^{z_i^{(1)}}, e^{z_i^{(2)}}, \ldots, e^{z_i^{(m)}})^T) = \sum_{i=1}^{n} \tilde{H}((z_i^{(1)}, z_i^{(2)}, \ldots, z_i^{(m)})^T) =$$

$$= \sum_{i=1}^{n} \tilde{H}(\underline{z}_i).$$

$\tilde{H}(\underline{z}_i)\underline{z}_i$-nek monoton nemcsökkenő konvex függvénye (hiszen monoton nemcsökkenő konvex függvény monoton nemcsökkenő konvex függvényéről van szó), vagyis $M(\underline{z}_i, \underline{z}_i + \underline{\delta}_i)$ megengedett ráforditásfüggvény. A megoldás az [1]-beli feladat megoldásával vektorkomponenenségenként megegyező, a tételben szereplő algoritmus értelemszerű átirása miatt.

2. A tételben a vektorok monotonitására, és az univerzális küszöbindexek létezésére vonatkozó viszonylag erős feltételeket gyengithetjük azáltal, hogy az $M(\underline{x}, \underline{y})$ megengedett ráforditásfüggvényt a következő lineáris kombinációként állithatjuk elő:

$$M(\underline{x}, \underline{y}) = \sum_{j=1}^{m} \lambda^{(j)} M(x^{(j)}, y^{(j)}) \qquad j = 1,2, \ldots, m$$

Ekkor ugyanis az optimumfeladat önálló, külön-külön megoldható egy-feltételes optimumfeladatokra bontható fel, a következő átalakithatóság miatt:

$$\sum_{i=1}^{n} M(\underline{x}_i, \underline{y}_i) = \sum_{i=1}^{n} \sum_{j=1}^{m} \lambda^{(j)} M(x_i^{(j)}, y_i^{(j)}) = \sum_{j=1}^{m} \lambda^{(j)} \left( \sum_{i=1}^{n} M(x_i^{(j)}, y_i^{(j)}) \right),$$

továbbá a feltételrendszer

$$(18) \qquad \sum_{i=1}^{n} y_i^{(j)} = y^{(j)} \quad \left(\text{vagy } \prod_{i=1}^{n} y_i^{(j)} = y^{(j)}\right) \qquad j = 1,2, \ldots, m$$

alakja miatt. Ezek a feladatok a következők:

$$\sum_{i=1}^{n} M(x_i^{(j)}, y_i^{(j)}) \to \min (j = 1,2, \ldots, m)$$

feltéve, hogy a (18) feltételek teljesülnek. Igy az $\{ x_i^{(j)} \}$ értékrendszerek minden $j = 1,2, \ldots, m$ esetén monoton sorozatba rendezhetők, és a küszöbindexek külön-külön határozandók meg az egyes részfeladatokra vonatkozóan. Ezek igy $k(j)$ alakúak lesznek $(j = 1,2, \ldots, m)$, és egymással nem kell feltétlenűl egyenlőeknek lenniük.

3. Az $M(\underline{x}, \underline{y})$ megengedett ráforditásfüggvényekre felirt matematikai modellben megfogalmazott minimumfeladatra tehát olyan megoldást találunk, amelynek algoritmikus megadásához nincs szükség a célfüggvény konkrét ismeretére, és általában teljesiti a bevezetőben már emlitett tulajdonságokat. Igy a gyakorlati felhasználás számára jól kezelhető, rá számitógépes program igen könnyen készithető.

Az általánositás további lépése lenne különböző $M_i(\underline{x}_i, \underline{y}_i)$ $i = 1,2, \ldots, n$ függvények valamint általánosabb tipusú feltételek melletti megoldás kidolgozása, amely azonban nem mindig látszik megvalósithatónak.

## Irodalom

[1] Dobó A., Szajcz S., A megbizhatóság növelésnek egy optimális elosztásáról. MTA III. Osztályának Közleményei. 1965. IV. kötet, 3.szám

[2] Dobó,A., Szajcz S., Optimális algoritmusok kialakitásáról. Matematika Alkalmazása 4. KGM ISZSZI Bp. 1971.

[3] Farkas Z., Munkaráforditások egy optimális elosztásáról. Diplomamunka. ELTE, 1967.

[4] Farkas Z., Erőforrások optimális szétosztása (Algoritmusok) Tanulmány. Készült a "Számitógépek alkalmazása" KGM – célprogram keretében, 1968-ban. 43. o.

[5] Farkas Z., Erőforrások optimális szétosztásáról Vállalatvezetés és vállalatszervezés 1. évf. (1969) 3.sz.

[6] Aczél J., Vorlesungen über Funktionalgleichungen und ihre Anwendungen. Stuttgart, 1962.

[7] Bellman, R., Dynamical Programming. Princeton Unversity Press Princeton, New Jersey, 1957.

[8] Karush, W., A General Algorithm for the Optimal Distribution of Effort. Management Science 9(1962) 1

[9] Saaty, T.A. Bram, J., Nonlinear Mathematics. Mc Graw-Hill Inc. 1964. 116.o.

# Р Е З Ю М Е

Обобщенное применение метода "Оптимизация с минимальной информацией" на проблемах распределений.

Золтан Фаркаш

В настоящей статье изложен один из результатов автора, связанный с методом "оптимизация с минимальной информацией". Это является решением следующей задачи:

$$\sum_{i=1}^{n} M(\underline{x}_i, \underline{y}_i) \to \min, \qquad \sum_{i=1}^{n} \underline{y}_i = \underline{y}, \ \underline{x}_1 \leqslant \underline{x}_2 \leqslant \ldots \leqslant \underline{x}_n,$$

$$\underline{x}_i, \underline{y}_i \in E^n, \quad \underline{x}_i \leqslant \underline{y}_i, \qquad i = 1, 2, \ldots, n;$$

и функция $M(\underline{x}, \underline{y})$ не известна только даны некоторые алгебраические свойства этой функции. Данный модел можно использовать в общих случаях. Например, модел можно интерпретировать обобщенной транспортной задачей, а алгоритм обобщенным решением некоторых проблем теории надежности /см. лит. [ 1 ]/.

## S u m m a r y

A generalized application of the method "Optimization with minimal information" to allocation problems

Zoltán Farkas

One of the author's mathematical result sonnected with the method "optimization by minimal information"is described. The problem solved is one of the generalized resource allocation problems as follows

$$\sum_{i=1}^{n} M(\underline{x}_i, \underline{y}_i) \to \min, \ \text{if} \ \sum_{i=1}^{n} \underline{y}_i = \underline{y}, \quad \text{where} \quad \underline{x}_1 \leqslant \underline{x}_2 \leqslant \ldots \leqslant \underline{x}_n,$$

$$\underline{x}_i, \underline{y}_i \in E^n, \ \underline{x}_i \leqslant \underline{y}_i, \quad i = 1, 2, \ldots, n;$$

and the objectiv function is not known, only well – defined properties are fulfilled by the function $M(\underline{x}, \underline{y})$. A general model and solution algorithm is given, which can be use under general conditions. Model can be interpret for instance as certain generalization of the transportation problem and algorithm gives a generalized solutions of some reliability theory problems (see e.g. ref. [1]).

MTA Számitástechnikai és Automatizálási Kutató Intézete, **Közlemények** 20/1978.

# A NEMLINEÁRIS PROGRAMOZÁS SZEKVENCIÁLIS MÓDSZEREI

Gerencsér László

A gazdaságos müszaki tervezés egyik hatékony módszere a matematikai programozás, ezen belül különösen a nemlineáris programozás. Az első nemlineáris programozási modellek megfogalmazását követően a vizsgálatok azok numerikus megoldása felé tolódtak el. Ezekről a módszerekről magyar nyelven kevés munka látott napvilágot. Mind a kutatómunka mind pedig az oktatómunka szempontjából szükségét érezzük annak, hogy újabb, különféle szintü igényeket kielégitő ismertetők szülessenek. Ebben az irányban születtek is eredmények ([5]). Egy újabb tanulmány is készült a Budapesti Müszaki Egyetem Továbbképző Intézetének megbizásából ([4]). Ez a jegyzet optimalizálási módszereket ismertet, 12 előadás anyagát. Ennek a jegyzetnek egy részét az alábbiakban részletesen ismertetjük. Az ismertetendő rész az igen fontos szerepet játszó szekvenciális módszerekkel foglalkozik.

Egyenlőtlenségfeltételekkel definiált nemlineáris programozási feladatok megoldására számos módszert dolgoztak ki. A módszerek először a már kidolgozott lineáris programozási kódok kiaknázására törekedtek. Ebbe a csoportba tartoznak az approximációs módszer, a megengedett irányok módszere és a metszősik módszer. Ezek a módszerek elsősorban gyengén nemlineáris feladatok megoldására alkalmasak, a változók száma azonban viszonylag nagy lehet. A módszerek egy másik csoportja nem támaszkodik a lineáris programozási kódokra, ehelyett a feladatot feltételnélküli optimalizálási feladatok egy sorozatával helyettesiti. Ezeket a módszereket összefoglalóan szekvenciális módszereknek nevezik. Ide tartoznak különböző büntetőfüggvény módszerek, a centrummódszer és a multiplikátormódszer. Előnyük, hogy alkalmasak erősen nemlineáris feladatok megoldására és hogy könnyen programozhatók. A megoldható feladatok mérete: 20-30 változó, illetve feltétel, ez müszaki jellegü feladatoknál általában elégséges is.

Most rátérünk az un. logaritmikus bünetetőfüggvényes módszer ismertetésére. Legyen adott a

(1) $\min f(x)$

(2) $g_i(x) \geqslant 0 \qquad i = 1, \ldots, m$

nemlineáris programozási probléma. Az egyenlőtlenségfeltételek által meghatározott $R$ tartományt egy $I(x)$ függvénnyel hozzuk kapcsolatba, amely a tartomány belsejében közel 0 értékü, a tartomány határához közeledve viszont az értéke tart plusz végtelenhez. Ilyen függvényt definiálhatunk pl. a következő képlettel:

(3) $$I(x) = -\sum_{i=1}^{m} \ln y_i(x).$$

Ez az un. logaritmikus büntetőfüggvény.

Vezessük még be a

(4) $P(x,r) = f(x) + rI(x) \quad r > 0$

segédfüggvényt, és tekintsük a

(5) $\min P(x,r)$

feltételnélküli minimalizálási feladat. Itt $r$ egy kicsiny pozitiv paraméter, A (5) feladat megoldása is $r$-től függ, jelöljük ezt $x(r)$-rel. Elég általános feltételek mellett megmutatható, hogy $x(r) \to x^*$ ha $r \to 0$, ahol $x^*$ a (1), (2) feladat megoldása. Az állitás pontosabb megfogalmazásától eltekintünk, ehelyett az eljárás lényegét kivánjuk egy egyszerü esetben megvilágitani. Nevezetesen tegyük fel, hogy a feladat konvex, vagyis az $f_1 - g_i$ függvények konvex függvények. Könnyü belátni, hogy ekkor $P(x,r)$ is konvex minden $r$-re. Ha még ezen túlmenően $P(x,r)$ szigoruan konvex, akkor $x(r)$ egyértelmüen meghatározott.

Tekintsük a

(6) $P_x(x(r)),r) = 0$

egyenletet. Hogy ezt részletesebben felirhassuk, számitsuk ki $I(x)$ gradiensét:

(7) $I_x(x) = -\sum_{i=1}^{m} g_{ix}(x)/g_i(x).$

Bevezetve az

(8) $\lambda_i(r) = r/g_i(x(r))$

jelölést a (11.6) egyenletből az

(9) $f_x(x(r)) - \sum_{i=1}^{m} \lambda_i(r) g_{ix}(x(r)) = 0$

egyenletet kapjuk. Ez az egyenlet a Kuhn-Tucker feltétel közelitésének takinthető. Ha $r$ tart 0-hoz, akkor a $\lambda_i(r)$ vagy 0-hoz tart, (ha $g_i(x(r))$ nagyobb mint egy pozitiv szám), vagy $\lambda_i(r) \to \lambda_i^*$ ahol $\lambda_i^*$ egy aktiv feltételhez tartozó Lagrange szorzó. Ez a heurisztikus gondolatmenet egzakt bizonyitássá fejleszthető, melynek eredményeképpen azt kapjuk, hogy $x(r) \to x^*_,$ $\lambda(r) \to \lambda^*$ ahol $x^*_,$ $\lambda^*$ kielégitik a Kuhn-Tucker feltételeket. Ez pedig konvex feladat esetén az optimalitás elégséges feltétele.

A vázolt gondolatmenetből az is kitünik, hogy az aktiv $i$ indexekre $g(x(r))$ nagyságrendje $r$ hacsak $\lambda_i^* \neq 0$. Ebből pedig általában következik, hogy $x^*$ és $x(r)$ eltérése is $r$ nagyságrendü. Másszóval $r$ csökkentésével a közelités pontossága lineárisan változik. További megszoritások mellett megmutatható, hogy az $x(r)$ görbe folytonosan differenciálható $r \geqslant 0$ esetén ($Itt$ $x(0) = x^*$). Ennek alapján az $x(r)$ görbék első vagy másodrendü Taylor-sorfejlesztéssel közelithetjük meg, s igy a konvergenciát extrapolációval gyorsithatjuk. Az extrapolációs számitások különösen egyszerüek, ha az egymásutáni $r$ értékeket az

(10) $\quad r_{k+1} = r_k / c$

képlet alapján választjuk, ahol $c$, szokásos érték 2 és 10 közé eső szám.

Lineáris extrapoláció esetén két egymáskövető $x^1 = x(r)$, $x^2 = x(r/c)$ ponton egy egyenest illesztünk, és az $x^3 = x(r/c^2)$ pontot közelitjük. Bevezetve a $q = 1/c$ jelölést, a lineáris extrapoláció az

(11) $\quad x^3 \approx -qx^1 + (1+q)x^2$

eredményt adja.

Ha a számitásokat nem akarjuk folytatni, akkor közvetlenül $x^*$ -t közelitjük. A lineáris extrapolációval ebben az esetben az

(12) $\quad x^* \approx (-qx^1 + x^2)/(1-q)$

eredményt kapjuk.

A $P(x,r)$ függvény minimalizálásakor egyik legnagyobb nehézség, hogy kicsiny $r$ esetén ez a függvény elfajult. Ezen azt értjük, hogy a szokásos függvényminimalizálási algoritmusok hátterét képező elméleti feltevések nem teljesülnek. Pontosabban arról van szó, hogy a $P(x,r)$ függvény Taylor-sorának részletösszegei a függvénynek rossz közelitései. A második deriváltakból alkotott Hesze-mátrix sajátértékei ugyanis eltérő nagyságrendüek. Az $r$ csökkentésével egyes sajátértékek véges értékhez tartanak, mások pedig végtelenhez tartanak. Ez utóbbi sajátértékek $r^{-1}$ nagyságrendüek. Ez az eredmény abból a megfontolásból adódik, hogy az $rP_{xx}(x(r),r)$ mátrixnak véges, de szinguláris határértéke van, nevezetesen:

(13) $\quad HH'$,

ahol

(14) $\quad H = (g_{1x}(x^*), \ldots, g_{px}(x^*))$

és $1, \ldots, p$ *az aktiv indexeket jelöli.*

A $P(x,r)$ függvény minimalizálására mégis gyakran használnak kvázi-Newton módszereket. Jó eredményt adank igen egyszerü direkt módszerek is,
A megbizhatóság növelésére vonatkozó vizsgálatok azonban még nem zárultak le.

A logaritmikus bünetetőfüggvény mellett számos más segédfüggvény jelent meg az irodalomban Az (1) (2) feladattal kapcsolatban szokás a

(15) $\quad P(x,r) = f(x) + r^{-1} \sum_{i=1}^{m} \{\min(0, g_i(x))\}^2$

u.n. négyzetes büntetőfüggvényt bevezetni. Ennek előnye a logaritmikus büntetőfüggvénnyel szemben az, hogy a számitás megkezdéséhez nem kell megengedett pontot keresni. Sőt a tipikus eset az, hogy $P(x,r)$ minimuma, $x(r)$ nem megengedett.

A (1), (2) feladat helyett tekintsünk most egy olyan feladatot, amelyben egyenlőtlenségfeltételek is szerepelnek:

(16) $\min f(x)$

(17) $g_i(x) \geqslant 0 \quad i = 1, \ldots, m$

(18) $h_j(x) = 0 \quad j = 1, \ldots, p.$

Az ilyen feladatokra vegyes büntetőfüggvényeket alkalmazunk. A logaritmikus és a négyzetes büntetőfüggvény együttes alkalmazásából a

(19) $$P(x,r) = f(x) - r\sum_{i=1}^{m} \ln g_i(x) + r^{-1} \sum_{j=1}^{p} h_j^2(x)$$

segédfüggvényt kapjuk.

Tiszta négyzetes büntetőfüggvényt pedig a

(20) $$P(x,r) = f(x) + r^{-1} \sum_{i=1}^{m} \{\min(0, g_i(x))\}^2 + r\sum_{j=1}^{p} h_j^2(x)$$

képlettel szerkeszthetünk. Mind a (15), (19), mind a (20) segédfüggvény esetén igaz marad az az észrevétel, hogy az $x(r)$ feltételnélküli és az $x^*$ optimum eltérése $r$ nagyságrendü.

## Irodalom

[1] K.H. Elster, Ch. Grossman: Nemlineáris optimumszámitási feladatok megoldása büntető és akadályfüggvényekkel. Megjelent a Közgazdasági operációkutatási alkalmazások c. kötetben. Közgazdasági és Jogi Könyvkiadó, 1976.

[2] A.V. Fiacco, G.P. Mc Cormick.: Nonlinear Programming: Sequential Unconstrained Minimization Techniques. Wiley, New York – London (1968).

[3] Gerencsér László: Nemlineáris programozási feladatok megoldása szekvenciális módszerekkel. Kandidátusi értekezés. Megjelent az MTA SzTAKI Tanulmányok sorozatban 49-es sorszámmal.

[4] Gerencsér László: Optimalizálás (jegyzet) Budapesti Müszaki Egyetem Továbbképző Intézete, (1977).

[5] Krekó Béla: Optimumszámitás (Nemlineáris programozás). Közgazdasági és Jogi Könyvkiadó, (1972).

[6] Rapcsák Tamás: Egy külsőpont eljárás konvex nemlineáris programozási feladatok megoldására. Alkalmazott Matematikai Lapok, 1(1975), 357-364.

[7] Rapcsák Tamás: A SUMT módszer alkalmazása logaritmikusan konkáv feltételi függvények esetén. MTA SzTAKI Közlemények, 2(1978).

## S u m m a r y

Sequential methods of nonlinear programming

László Gerencsér

Description and basic properties of SUMT methods are presented.

## Р Е З Ю М Е

Последовательные методы нелинейного программирования

Ласло Геренчер

Описывается метод и его основные свойства.

789668 MTA KESZ Sokszorosító. F. v.: dr. Héczey Lászlóné